A Theory of Resolution

Leo Bachmair
Harald Ganzinger

MPI–I–97–2–005 April 1997

Authors' Addresses

Leo Bachmair
Department of Computer Science
SUNY at Stony Brook
Stony Brook, NY 11794, U.S.A.
`leo@cs.sunysb.edu`
`www.cs.sunysb.edu/~leo`

Harald Ganzinger
Max-Planck-Institut für Informatik
Im Stadtwald
D-66123 Saarbrücken, Germany
`hg@mpi-sb.mpg.de`
`www.mpi-sb.mpg.de/~hg`

Publication Notes

This paper is a preliminary version of an article intended for publication in the Handbook of Automated Reasoning, edited by J.A. Robinson and A. Voronkov.

Acknowledgements

This research was supported in part by the National Science Foundation under grants INT-9314412 and CCR-9510072, by the German Science Foundation (Deutsche Forschungsgemeinschaft) under grants Ga 261/4-1 and Ga 261/7-1, and by the ESPRIT Basic Research Working Group 22457 (Construction of Computational Logics II).

Abstract

We review the fundamental resolution-based methods for first-order theorem proving and present them in a uniform framework. We show that these calculi can be viewed as specializations of non-clausal resolution with simplification. Simplification techniques are justified with the help of a rather general notion of redundancy for inferences. As simplification and other techniques for the elimination of redundancy are indispensable for an acceptable behaviour of any practical theorem prover this work is the first uniform treatment of resolution-like techniques in which the avoidance of redundant computations attains the attention it deserves. In many cases our presentation of a resolution method will indicate new ways of how to improve the method over what was known previously. We also give answers to several open problems in the area.

Keywords

Automated Theorem Proving, Resolution, Simplification, Saturation

Contents

1 Introduction

Resolution is one of the main computational methods in logic programming and automated theorem proving. The purpose of this paper is to describe a general theory of resolution and to show how it can be used to explain the basic properties of most known resolution-based deduction methods.

The basis for the theory is a non-clausal resolution rule that incorporates various restrictions imposed by ordering constraints and selection functions and also limits the extent of implicit factoring. Orderings and selection functions may be chosen from large classes of orderings and selection functions, respectively, and each specific setting of these parameters results in a specific calculus. The second main component of the theory is an abstract concept of redundancy, based on which most techniques for simplifying or deleting formulas can be formulated. The fundamental result of the paper is the refutational completeness of these calculi and its compatibility with the redundancy criterion. The completeness result applies to any combination of deductive inferences with redundancy-based simplification mechanisms. The meaning of "any combination" is made precise by a formal notion of fair theorem proving derivations together with the concept of a saturated matrix. Fairness will guarantee that limit systems of such derivations are saturated up to redundancy.

In the second part, in Sections 7 and 8, the paper reformulates the major resolution-based methods of automated deduction within the given framework. We will show that these methods are specific instances of the theory and can be obtained from appropriate settings of its parameters. This analysis provides for more than just reestablishing known results in a new framework. In some cases (e.g., the inverse method) the exact relationship to standard resolution was not known before. In each case of a method that we consider we will be able to give a precise account of which simplification techniques are compatible with a particular inference system. For instance we will answer open questions about certain variants of hyper-resolution and boolean ring based resolution and superposition and extend known results about ordered theory resolution.

The reduction of any particular such strategy to the general case will be straightforward without requiring any major technical overhead. That is in sharp contrast to many papers in the literature where some specific variation of the resolution scheme is often accompanied by a very technical, lengthy and specific completeness argument providing little, if any, insight into the problem itself. In this paper we shall show that only a few basic principles suffice to explain resolution in general.

The theory laid down in this paper does not require formulas to be in any normal form. (We do assume, however, that quantifiers have been eliminated by skolemization.) Computing with normal forms admits an efficient implementation of search and simplification. In the paper we shall, among others, specialize non-clausal resolution to clausal normal form, to Maslov's "super clauses" (disjunctions of conjunctions of literals), and to sums of products in the Boolean ring. In our approach transformation to normalform need not be performed as a separate preprocessing but can be intertwined with inference computation in an arbitrary manner. Among others, this provides ways of dealing with logical equivalence. For instance, in the clausal case equivalences need not initially be converted into conjunctions of implications, as this may exponentially increase the size of the original formula. While there are translation schemes that avoid such an exponential increase in the size of formulas, the translation also destroys much of the structure of formula. Equivalences can often be oriented into rewrite rules that transform formulas into a normal form which is well-suited for the particular application at hand (cf. Section 6.4).

In Section 10 we shall discuss the rôle of resolution-based methods, not only for refutational theorem proving, but also, and especially so, as a tool for analyzing and compiling presentations of logical theories. We will indicate how the resolution theory can be applied to obtain refinements of tableau-based theorem proving methods. We will argue that the notion of a "closed tableau" should be generalized to that of a "saturated tableau" in which all paths are saturated, up to redundancy, by ordered resolution. It will be briefly explained how saturation may help in automatically generating decision procedures for a theory so that a certain complexity bound can be guaranteed. Finally we indicate in which way saturation may be a mechanical tool for generating variants of resolution calculi that are specifically tailored to certain theories such as orderings or congruence. This will shed some new light on how to compute with large, but structured theories.

2 Preliminaries

2.1 Formulas and Clauses

We consider quantifier-free first-order formulas built from variables, function symbols, predicate symbols and logical connectives. We will deal with the logical symbols $\top$ (verum), $\bot$ (falsum), $\neg$ (negation), $\vee$ (disjunction), $\wedge$ (conjunction), $\rightarrow$ (implication), $\oplus$ (exclusive disjunction), and $\leftrightarrow$ (equivalence), though our results apply to other connectives as well. A *term* is either a variable or an expression $f(t_1, \ldots, t_n)$, where f is an n-ary function symbol and $t_1, \ldots, t_n$ are terms. For the major part of this paper we will only be concerned with *ground expressions* in which terms do not contain any variables.[1] Hence, unless not stated otherwise, we shall always implicitly assume that terms are ground, and we will speak of *first-order terms*

[1] The lifting problem, that is, the problem of extending theorem proving strategies and associated completeness results to formulas with variables will only briefly be discussed in Section 9.

or *first-order expressions* to explicitly admit the presence of variables. An *atomic formula* (or *atom*) is an expression $P(t_1, \ldots, t_n)$, where P is a predicate symbol of arity n and $t_1, \ldots, t_n$ are terms. A predicate symbol with arity 0 is called a *propositional constant*. A *literal* is an expression A (a *positive* literal) or $\neg A$ (a *negative* literal), where A is an atomic formula. The two literals A and $\neg A$ are said to be *complementary*. The letters F and G will be used to denote formulas.

Calculi for automated deduction are often described in terms of constructs that represent formulas, but abstract from inessential aspects of the syntax or encode additional structural information. For example, multiple disjunctions or conjunctions may be conveniently represented as sequences (or multisets), due to the associativity (and commutativity) property of the connective.

A *multiset* over a set S is a function Σ from S to the natural numbers. Intuitively, $\Sigma(x)$ specifies the number of occurrences of x in Σ. We say that x is an *element* of Σ if $\Sigma(x) > 0$. A set may be thought of as a multiset Σ for which $\Sigma(x)$ is 0 or 1, for all x. A multiset Σ is *finite* if $\Sigma(x) = 0$ for all but finitely many x. The *union* and *intersection* of multisets are defined by the identities $\Sigma_1 \cup \Sigma_2(x) = \Sigma_1(x) + \Sigma_2(x)$ and $\Sigma_1 \cap \Sigma_2(x) = \min(\Sigma_1(x), \Sigma_2(x))$. If Σ is a multiset and S a set, we write $\Sigma \subseteq S$ to indicate that every element of (the multiset) Σ is an element of (the set) S, and use $\Sigma \setminus S$ to denote the multiset Σ' for which $\Sigma'(x) = 0$ for any x in S, and $\Sigma'(x) = \Sigma(x)$, otherwise. We often use sequences or set-like notation to denote multisets. For instance, if Σ and Δ are multisets, we write Σ, Δ instead of $\Sigma \cup \Delta$, and Σ, A instead of $\Sigma \cup \{A\}$. For example, by $\neg A, B, B$ we denote the multiset Σ over formulas for which $\Sigma(\neg A) = 1$, $\Sigma(B) = 2$, and $\Sigma(F) = 0$, for all other formulas F.

Depending on the context, finite multisets of formulas either denote the disjunction or the conjunction, respectively, of their elements. In case it denotes a disjunction we call the multiset a *general clause*. The empty clause represents the constant $\perp$. If $(F_1, \ldots, F_n)$ is a clause, by $\neg(F_1, \ldots, F_n)$ we denote the formula $\neg F_1 \wedge \ldots \wedge \neg F_n$. A *standard clause* is a clause in which all elements are literals. For standard clauses we also use the notation $L_1 \vee L_2 \vee \ldots \vee L_n$ as in this case the distinction between a disjunction and a multiset is inessential. A clause is called a *Horn clause* if it is a standard clause which has exactly one positive literal, called the *head* of the clause, or if it is of the form $\neg A_1 \vee \ldots \vee \neg A_n \vee \perp$, with atoms A_i. (The latter form is our notation for Horn clauses with an *empty head*. In particular $\perp$ denotes the empty Horn clause.) We will use greek letters Γ, Δ, Σ to denote general clauses. Standard clauses will be denoted by C and D. Multisets of formulas that represent the conjunction of their elements are called *(general) dual clauses*. For dual clauses we obtain the corresponding notions and notations by replacing the symbols $\vee$ and $\perp$ by their duals $\wedge$ and $\top$, respectively.

We write $E[E']$ to indicate that the expression E contains E' as a subexpression; and denote by $E[E'']$ the result of replacing in E an indicated occurrence of E' by E''. By $E[E'/E'']$ we denote the result of simultaneously replacing *all* occurrences of E' by E''. We will also have to consider partial substitutions for subexpressions. If E' is a subexpression of E, by $E[E'|E'']$ we denote any expression that differs from E only in that some *(at least one)* of the occurrences of E' in E have been replaced by E''.

2.2 Herbrand Interpretations

A (*Herbrand*) *interpretation* is a set of ground atoms. A ground atom A is said to be *true* in a Herbrand interpretation I if $A \in I$, and *false* otherwise. The logical symbols are interpreted in the usual way: The constant $\top$ is true in all interpretations, whereas $\bot$ is false in all interpretations. A conjunction $A \wedge B$ is true in I if, and only if, both A and B are true in I; a disjunction $A \vee B$ is true if at least one of A and B is true; etc. The truth value of a formula depends only on the truth values assigned to its atomic formulas. A clause $(F_1, \ldots, F_n)$ is true in I if one of the formulas F_i is true in I.

An interpretation I is called a *model* of E if E is true in I. I is a model of a multiset of expressions N, if I is a model of any expression in N. An expression or a multiset of expressions is called *satisfiable* or *consistent* if it has a model; and *unsatisfiable* or *inconsistent* otherwise. An expression that is true in all interpretations is called *valid*, or a *tautology.* We also say that E' is a *logical consequence* of an expression or a multiset of expressions E (or *logically follow*s from E), written $E \models E'$, if E' is true in all models of E. Two expressions E and E'' are said to be *(logically) equivalent*, written $E \equiv E''$, if, and only if, E and E' have the same truth value in each interpretation I. By a *contradiction* we mean an inconsistent expression that contains only the constants $\top$ and $\bot$ and logical connectives, but no function or predicate symbols. For example, $\bot$ and $\top \supset \bot$ are contradictions. The formula $A \wedge \neg A$ is inconsistent, but not a contradiction in our sense.

Theorem provers are procedures that check whether a given expression E (the "goal") is a logical consequence of a multiset of expressions N (the "theory"). Refutational theorem provers deal with the equivalent problem of showing that $N \cup \{\neg E\}$ is inconsistent by deriving a contradiction from $N \cup \{\neg E\}$.

2.3 Rewrite Systems

Rewrite systems are a basic tool for describing a variety of theorem proving techniques. We will primarily use rewrite systems for rewriting formulas. We use the letters $\alpha, \beta, \ldots$ to denote metavariables ranging over formulas. Syntactically, these variables are treated like propositional constants.

A *substitution* is a mapping from variables to formulas. By $E\sigma$ we denote the result of applying the substitution σ to a formula E and call $E\sigma$ an *instance* of E. If $E\sigma$ is *ground* (i.e., contains no variables), we speak of a *ground instance* of E. Composition of substitutions is denoted by juxtaposition. Thus, if τ and ρ are substitutions, then $x\tau\rho = (x\tau)\rho$, for all variables x.

An *equivalence (relation)* is a reflexive, transitive, symmetric binary relation. For example, logical equivalence is indeed an equivalence relation. A binary relation $\Rightarrow$ on formulas (with variables) is called a *rewrite relation* if $F' \Rightarrow F''$ implies $F[F'] \Rightarrow F[F'']$, for all formulas F, F' and F''. If $\Rightarrow$ is a binary relation, we denote by $\Rightarrow^+$ its transitive closure; by $\Rightarrow^*$ its transitive-reflexive closure; by $\Leftrightarrow$ its symmetric closure; and by $\Leftrightarrow^*$ its transitive-reflexive-symmetric closure.

A *rewrite system* is a binary relation on formulas with metavariables, the elements of which are called *rewrite rules* and written $F \Rightarrow F'$. (We occasionally speak of a *two-way rewrite rule*, and write $F \Leftrightarrow F'$, if a rewrite system contains both $F \Rightarrow F'$ and $F' \Rightarrow F$.) If R is a rewrite system, we denote by $\Rightarrow_R$ the smallest rewrite relation that contains all instances $F\sigma \Rightarrow F'\sigma$ of rules in R. We also say that F can be *rewritten to* F' by R, if $F \Rightarrow_R F'$. We extend the notion of rewriting to multisets by defining: $\Sigma \Rightarrow_R \Sigma'$ if Σ can be written as Δ, F and Σ' as Δ, F', for some clause Δ and formulas F and F' with $F \Rightarrow_R F'$. Thus, a rewrite relation defined on formulas can be extended to clauses, as well as to matrices and sequents that we shall introduce in Section 4.1 below.

Expressions that can not be rewritten are said to be in *normal form.* We write $F \Rightarrow^!_R F'$ to indicate that $F \Rightarrow^+_R F'$ and F' is in normal form. We say that R *terminates* if there is no infinite sequence $F_0 \Rightarrow_R F_1 \Rightarrow_R \cdots$ of rewrite steps. If R terminates, then every formula can be rewritten to a normal form (in zero or more steps).

If R and S are rewrite systems, we denote by R/S (R *modulo* S) the rewrite system consisting of all rules $F \Rightarrow F'$, such that $F \Leftrightarrow^*_S G \Rightarrow_R G' \Leftrightarrow^*_S F'$, for some formulas G and G'.

2.4 Orderings

A (strict) *partial ordering* is a transitive and irreflexive binary relation; a *quasi-ordering* a reflexive and transitive binary relation. The reflexive closure of a strict ordering is a quasi-ordering. On the other hand, if $\succeq$ is a quasi-ordering, then its *strict part* $\succ$ defined by: $x \succ y$ if $x \succeq y$ but not $y \succeq x$, is a strict ordering. We may also define an equivalence relation by: $x \sim y$ if $x \succeq y$ and $y \succeq x$. We say that an ordering $\succ'$ *extends* $\succ$ if the latter is a subset of the former (i.e., $x \succ' y$ whenever $x \succ y$).

For example, we may compare formulas by their size and define either a strict ordering: $F \succ G$ if G is shorter (as a string) than F; or a quasi-ordering: $F \succeq G$ if F is not shorter than G. If F and G are of the same length, we have $F \succeq G$ and $G \succeq F$, but neither $F \succ G$ nor $G \succ F$. This ordering extends the subformula ordering. The example also shows that the reflexive closure of the strict part of a quasi-ordering may be different from the quasi-ordering.

A strict ordering $\succ$ is said to be *well-founded* if there is no infinite sequence $x_1 \succ x_2 \succ \cdots$ of elements. A quasi-ordering is well-founded if its strict part is well-founded. A strict ordering is said to be *total* (on S) if for any two distinct elements x and y (in S) we have either $x \succ y$ or $y \succ x$.

We say that an ordering $\succ$ has the *subterm property* if $E[E'] \succeq E'$, for all expressions E and E'. A *rewrite ordering* is an ordering that is also a rewrite relation; a *reduction ordering*, a well-founded rewrite ordering; and a *simplification ordering*, a reduction ordering with the subterm property. Note that a rewrite system R terminates if, and only if, there exists a reduction ordering $\succ$ such that $E\sigma \succ E'\sigma$, for each rule $E \Rightarrow E'$ in R and each substitution σ.

In this paper, we mainly use orderings defined with respect to the tree structure of terms and formulas. Let $\succ$ be an ordering, called a *precedence*, on the given set of (function, predicate and logical) symbols. The corresponding *lexicographic path ordering* $\succ_{lpo}$ is defined by:

$s = f(s_1, \ldots, s_m) \succ_{lpo} g(t_1, \ldots, t_n) = t$ if and only if

(i) $f \succ g$ and $s \succ_{lpo} t_i$, for all i with $1 \leq i \leq n$; or

(ii) $f = g$ and, for some j, we have $(s_1, \ldots, s_{j-1}) = (t_1, \ldots, t_{j-1})$, $s_j \succ_{lpo} t_j$, and $s \succ_{lpo} t_k$, for all k with $j < k \leq n$; or

(iii) $s_j \succeq_{lpo} t$, for some j with $1 \leq j \leq m$.

If the precedence is well-founded, the lexicographic path ordering is a simplification ordering. It is total on ground formulas whenever the given precedence is total.

Any ordering on a set S can be extended to an ordering $\succ_{mul}$ on finite multisets over S as follows: $\Sigma \succ_{mul} \Delta$ if (i) $\Sigma \neq \Delta$ and (ii) whenever $\Delta(x) > \Sigma(x)$ then $\Sigma(y) > \Delta(y)$, for some y such that $y \succ x$. (Here $>$ denotes the standard "greater-than" relation on the natural numbers.) Given a multiset, a smaller multiset is obtained by replacing an element by zero or more occurrences of smaller elements. If an ordering $\succ$ is total (resp., well-founded), so is its multiset extension. For example, if we order formulas by their size, then $P(f(a)) \succ Q(a)$ and $\{P(f(a))\} \succ_{mul} \{P(a), Q(a)\}$. For simplicity we usually use the same symbol to denote both an ordering and its multiset extension.

For a survey on termination orderings, see (Dershowitz 1987).

A particular class of reduction orderings on formulas, called admissible orderings, will be of importance below. A reduction ordering $\succ$ on formulas is called *admissible* if (i) $A \succ \top$ and $A \succ \bot$, for all atoms A; and (ii) if for all atoms B in G there exists an atom A in F such that $A \succ B$ then $F \succ G$. An ordering $\succ$ on clauses is *admissible* if it is the multiset extension of an admissible ordering on formulas.

For instance, any lexicographic path ordering is admissible if predicate symbols have higher precedence than logical symbols, and $\top$ and $\bot$ are lowest in precedence.

3 Deduction

Theorem provers are first of all deductive systems. In a refutational theorem prover new formulas are deduced from given ones with the goal of obtaining a contradiction. The most widely used inference rule for that purpose is resolution, which was originally introduced by Robinson (1965*b*). We will present a general version of resolution that applies to arbitrary clauses, of which the more common version of resolution that applies to standard clauses only is a special case.

3.1 General Resolution

For our purposes, an *inference rule* is an n-ary relation on expressions, where $n \geq 1$. The elements of such a relation are usually written as

$$\frac{E_1 \ldots E_{n-1}}{E}$$

and called *inferences*. The expressions $E_1, \ldots, E_{n-1}$ are called the *premises*, and E the *conclusion*, of the inference. We also speak of an inference *from* N if all premises are elements of N. An *inference system* is a collection of inference rules.

An inference is *sound* if the conclusion is a logical consequence of the premises, i.e., $E_1, \ldots, E_{n-1} \models E$. The following definition of resolution for formulas is sound.

General resolution:

$$\frac{F[G] \quad F'[G]}{F[G/\bot] \vee F'[G/\top]}$$

We speak of a resolution *on* G and call the conclusion of the inference a *resolvent* of the two premises. We also call F the *positive*, F' the *negative* premise, and G the *resolved subformula*.

Resolution is a sound inference. In fact, suppose that I is an interpretation in which both premises are valid. In I, the propositional formula G

is either true or false. If G $[\neg G]$ is true in I, so is $F[G/\top]$ $[F[G/\bot]]$, and hence the resolvent.

In the following example we derive a contradiction from three given input formulas by general resolution.

$$
\begin{array}{lll}
(1) & A \leftrightarrow B & \text{[input]} \\
(2) & \neg A \vee \neg B & \text{[input]} \\
(3) & A \vee B & \text{[input]} \\
(4) & (\bot \vee B) \vee (\top \leftrightarrow B) & \text{[resolving on } A \text{ in (3) and (1)]} \\
(5) & (\bot \leftrightarrow B) \vee (\neg\top \vee \neg B) & \text{[resolving on } A \text{ in (1) and (2)]} \\
(6) & (\bot \vee \bot) \vee (\top \leftrightarrow \bot) \vee (\bot \leftrightarrow \top) \vee (\neg\top \vee \neg\top) & \\
 & & \text{[resolving on } B \text{ in (4) and (5)]}
\end{array}
$$

(6) is a contradiction as each disjunct simplifies to $\bot$. Hence, by the soundness of the inference, there cannot be any interpretation in which the three input formulas are simultaneously valid.

A particular form of resolution is *self-resolution* in which the two premises are the same formula. Given a finite set of input formulas to be refuted, one might as well simply iterate self-resolution on their conjunction until all atoms habe been resolved:

$$
\begin{array}{lll}
(1) & (A \leftrightarrow B) \wedge (\neg A \vee \neg B) \wedge (A \vee B) & \text{[input]} \\
(2) & [(\bot \leftrightarrow B) \wedge (\neg\bot \vee \neg B) \wedge (\bot \vee B)] & \\
 & \vee \;\; [(\top \leftrightarrow B) \wedge (\neg\top \vee \neg B) \wedge (\top \vee B)] & \text{[resolving on } A \text{ in (1)]} \\
(3) & [(\bot \leftrightarrow \bot) \wedge (\neg\bot \vee \neg\bot) \wedge (\bot \vee \bot)] & \\
 & \vee \;\; [(\top \leftrightarrow \bot) \wedge (\neg\top \vee \neg\bot) \wedge (\top \vee \bot)] & \\
 & \vee \;\; [(\bot \leftrightarrow \top) \wedge (\neg\bot \vee \neg\top) \wedge (\bot \vee \top)] & \\
 & \vee \;\; [(\top \leftrightarrow \top) \wedge (\neg\top \vee \neg\top) \wedge (\top \vee \top)] & \text{[resolving on } B \text{ in (2)]}
\end{array}
$$

(3) is a contradiction as each disjunct contains a false conjunct.

These two examples indicate the wide spectrum of possible resolution strategies, ranging from local strategies where replacement of subformulas — atoms, in fact — is confined to single clauses, to global ones in which the entire state of the theorem proving process is modified nonlocally at each step.

Note that the order among the two premises is significant. For example, resolution on A in $A \vee B$ and $A \supset C$ yields the resolvent $(\bot \vee B) \vee (\top \supset C)$, which is logically equivalent to $B \vee C$. If we exchange the premises, we obtain $(\bot \supset C) \vee (\top \vee B)$, a tautology.

Non-clausal resolution is *refutationally complete* in that a contradiction can be deduced from any inconsistent set of formulas N. As the previous example indicates, if N is finite, one may replace N by the conjunction of its elements and then apply self-resolution until all atoms are eliminated. Since a self-resolvent of a formula is satisfiable if and only if the formula is, the final resolvent is an explicit representation of satisfiablity or unsatisfiablity of the initial N. This strategy resembles the Davis-Putnam procedure that we describe in more detail in Section 10.1. In the context of first-order logic one might in fact restrict attention to finite N. This is a consequence of the compactness theorem for first-order logic. However, a proof procedure that would exploit compactness by nondeterministically guessing a finite and inconsistent set of ground instances of a given inconsistent set of first-order formulas, and then apply self-resolution to verify the inconsistency, does not appear to be of much interest in practice. The main problem in the non-propositional case is not to demonstrate the inconsistency of a propositional instance but rather to "efficiently" identify those instances that are inconsistent. Hence it is important to see that resolution is refutationally complete also for *infinite* sets of formulas. Let us call a (possibly infinite) set of formulas N *saturated* with respect to an inference system $\mathcal{J}$ if the conclusion of any inference in $\mathcal{J}$ from premises in N is contained in N.

We prove the refutational completeness of general resolution by showing that any inconsistent saturated set of formulas contains a contradiction. More specifically, we will show that any set that is saturated under general resolution and contains no contradiction, has a model. For that purpose we use the following construction of a Herbrand interpretation by induction based on admissible orderings on formulas.

Let $\succ$ be a total admissible ordering on formulas. (In Section 2.4 we have argued that many such orderings can be constructed with the concept of lexicographic path orderings.) Given a set of formulas N, we use induction with respect to $\succ$ to define a Herbrand interpretation I_F and a set ε_F, for each formula F in N, as follows.

Definition 3.1 *Let us assume, as an induction hypothesis, that ε_G are defined for all formulas G in N with $F \succ G$. Then let I_F be the set $\bigcup_{F \succ G} \varepsilon_G$. Furthermore, if A is the maximal atomic formula in F, then $\varepsilon_F = \{A\}$ if (i) $A \notin I_F$ and (ii) F is false in I_F, but F is true in $I_F \cup \{A\}$. Otherwise, ε_F is the empty set.*

We also say that F *produces* A, and call F a *productive formula*, if $\varepsilon_F = \{A\}$. Finally, by $I_N^{\succ}$, or simply I_N, we denote the Herbrand interpretation $\bigcup_{F \in N} \varepsilon_F$.

The construction is designed to render the formulas of N true in I_N. The interpretation I_F is intended to be a model of the set N_F of those formulas in N that are smaller than F; and ε_F is meant to be a minimal extension of I_F that makes F true.

Example 3.2 We assume an atom ordering $E' \succ E \succ D \succ C \succ B \succ A$. The following table depicts the model construction for a set of formulas which the table is assumed to list in ascending order.

formula F	interpretation I_F	ε_F	remarks
$A \vee B$	$\emptyset$	$\{B\}$	B is maximal
$B \vee C$	{B}	$\emptyset$	F is true in I_F
$B \to C$	{B}	$\{C\}$	
$B \to (C \wedge E) \vee D$	$\{B, C\}$	$\{E\}$	
$B \wedge E \to D$	$\{B, C, E\}$	$\emptyset$	D not maximal
$E \wedge E'$	$\{B, C, E\}$	$\{E'\}$	

The construction will not always yield a model of N. In the example, the generated interpretation is $\{B, C, E, E'\}$, and the second last formula is not satisfied in it. But we have the following lemmas:

Lemma 3.3 *If $F \in N$ is productive, then F is true in I_N.*

Proof. By the construction, if F produces an atom A then F is true in $I_F \cup \{A\}$. As none of the atoms in $I_N \setminus (I_F \cup \{A\})$ occur in F (these must be larger than A which itself is the maximal atom in F), F remains true in I_N. □

The question is whether non-productive formulas are true in I_N. For saturated sets N this is indeed the case.

Theorem 3.4 *If N is saturated and contains no contradiction, then I_N is a model of N.*

Proof. For simplicity let us denote I_N by I. By the above lemma, productive formulas are true in I. Let F be the smallest non-productive formula that is false in I (the smallest "counterexample," so to say). Since N contains no contradiction, F must contain a non-logical symbol. Let A be the maximal atom in F. We consider two cases.

(i) Suppose $A \notin I$. Since F is false in I, the formula $F[A/\bot]$ is also false in I. But the truth value of any atom in $F[A/\bot]$ is the same in I_F as in I and therefore $F[A/\bot]$ is false in I_F as well. But then $F[A/\top]$ must also be false in I_F, for otherwise F would be productive. The resolvent $F[A/\bot] \vee F[A/\top]$ is false in I_F and in I. By saturation, it is contained in N. Since it contains only symbols strictly smaller than A in the given ordering, it is smaller than F. But this contradicts our assumption that F is the smallest false formula in N.

(ii) Suppose $A \in I$. Let F' be the formula producing A. Thus, $F'[A/\bot]$ is false in both $I_{F'}$ and I. By assumption, $F[A/\top]$ is false in I. By saturation, the non-clausal resolvent $F'[A/\bot] \vee F[A/\top]$ is contained in N. This resolvent is false in I, yet smaller than F, which is a contradiction.

In sum, we have proved that none of the formulas in N is false in I_N. In other words, I_N is a model of N. □

The idea underlying this completeness proof is simple. We show that if a minimal counterexample is not a contradiction, then an even smaller counterexample can be deduced by resolution. Resolving on minimal counterexamples is an effective and deterministic stragegy for saturating finite sets of formulas until either a contradiction is found, or else I_N yields a (minimal) model of N.

Example 3.5 Continuing the Example 3.2 we observe that $B \wedge E \to D$ is the smallest false formula and has E as maximal atom. E is produced by $B \to (C \wedge E) \vee D$. Resolving these two formulas, according to case (ii) of the proof, yields $(B \to (C \wedge \bot) \vee D) \vee (B \wedge \top \to D)$. Adding this resolvent and redoing the model construction yields

F	I_F	ε_F	remarks
$A \vee B$	$\emptyset$	$\{B\}$	
$B \vee C$	{B}	$\emptyset$	
$B \to C$	{B}	$\{C\}$	
$(B \to (C \wedge \bot) \vee D) \vee (B \wedge \top \to D)$	$\{B, C\}$	$\{D\}$	
$B \to (C \wedge E) \vee D$	$\{B, C, D\}$	$\emptyset$	true in I_F
$B \wedge E \to D$	$\{B, C, D\}$	$\emptyset$	true in I_F
$E \wedge E'$	$\{B, C, D\}$	$\emptyset$	

Since E is no longer produced, $E \wedge E'$ cannot be made true by producing E'. $E \wedge E'$ remains the smallest false formula, and case (i) in the proof of the Theorem applies. Self-resolving E' in $E \wedge E'$ produces $(E \wedge \bot) \vee (E \wedge \top)$. Now the construction proceeds as

F	I_F	ε_F
$A \vee B$	$\emptyset$	$\{B\}$
$B \vee C$	{B}	$\emptyset$
$B \to C$	{B}	$\{C\}$
$(B \to (C \wedge \bot) \vee D) \vee (B \wedge \top \to D)$	$\{B, C\}$	$\{D\}$
$(E \wedge \bot) \vee (E \wedge \top)$	$\{B, C, D\}$	$\{E\}$
$B \to (C \wedge E) \vee D$	$\{B, C, D, E\}$	$\emptyset$
$B \wedge E \to D$	$\{B, C, D, E\}$	$\emptyset$
$E \wedge E'$	$\{B, C, D, E\}$	$\{E'\}$

and results in the model $\{B, C, D, E, E'\}$

Thus, a saturated set either contains a contradiction or else has a model. Another consequence of the Theorem 3.4 is that propositional logic is compact. A finite or denumerably infinite set N of propositional formulas can be saturated under general resolution in at most ω steps. If it is inconsistent the saturation contains a contradiction. This contradiction has a finite resolution proof involving only finitely many formulas in N.

General resolution, as a sound and complete inference rule, provides a suitable deductive base for refutational theorem proving. For practical purposes it is too general, though, in that too many formulas can be deduced (the "search space" is too large). We next discuss useful restrictions on resolution that do not impair its completeness.

3.2 Selection Functions and Ordering Restrictions

An inspection of the proof of Theorem 3.4 reveals that we have actually established a stronger result than is stated in the theorem. The proof only requires resolution on atomic formulas; in fact, only on the maximal atom in a ground formula. In other words, we may impose certain ordering restrictions on resolution. Sometimes, resolution on non-maximal subformulas will be useful, though, and we propose selection functions as a corresponding control mechanism. Furthermore, we will also relax the requirement that *all* occurrences of a resolved formula be replaced. The refined inference rules have to be formulated in terms of clauses, as it is necessary to distinguish (sub)formulas introduced by an inference from original input formulas.

By a *selection function* we mean a mapping S that assigns to each general clause Σ a (possibly empty) set $S(\Sigma)$ of nonempty sequences of (distinct) atoms in Σ such that either (i) $S(\Sigma)$ is empty or else (ii) for any interpretation I in which Σ is false there exists a sequence $A_1, \ldots, A_k$ in $S(\Sigma)$ such that A_j is true in I, for all $1 \leq j \leq k$. If $A_1, \ldots, A_k$ is in $S(\Sigma)$ we say that the sequence $A_1, \ldots, A_k$ *is selected in* Σ *(by* S*)*. If a sequence in $S(\Sigma)$ consists of a single atom A, we call A a *selected atom in* Σ. By selecting sequences of more than one atom one specifies their simultaneous replacement in a resolution inference. We say that Σ contains *no selected atom*, whenever $S(\Sigma)$ is empty.

For example, consider $\Sigma = (\neg A \wedge \neg B, \neg A', B')$. Then both $S(\Sigma) = \{(A, A'), (B, A')\}$ and $S(\Sigma) = \{A, B\}$ satisfy (ii). A third possible choice for selection would be $S(\Sigma) = \{A'\}$.

In general, verifying (ii) for any given S and clause Σ may be a computationally hard problem. Certain syntactic criteria are helpful in this regard. For instance, if C is a simple clause, then those atoms that occur in negative literals of C are precisely the ones that may be selected, and any set of sequences of negative atoms forms a legal selection. For general clauses, a suitable generalization of the notion of *polarity* of a formula in an expression is useful. A subformula of F' in $E[F']$ is said to be *positive* (resp., *negative*) if $E[F'/\top]$ (resp., $E[F'/\bot]$) is a tautology. In that case F' (resp., $\neg F'$) implies E.

For example, in a disjunction $A \vee B$ both A and B are positive, whereas in a conjunction $A \wedge B$ the two subformulas A and B are neither positive nor negative. A subformula may occur both positively and negatively (e.g., A in $A \vee \neg A$ or $A \equiv A$), in which case the formula is a tautology. For simple clauses $\neg A_1 \vee \ldots \vee \neg A_m \vee B_1 \vee \ldots \vee B_n$ we obtain the usual notion of polarity in that all atoms A_i are negative and all atoms B_j are positive. With this notion of polarity for general clauses, any negative atom A in a clause may be selected since a negative A cannot be false in any interpretation in which the clause is false.

In general, determining whether an atom A is positive or negative in E requires one to check whether $E[A/\top]$ or $E[A/\top]$ is a tautology, which again is a computationally hard problem. We may employ sufficient syntactic criteria that allow us to identify certain positive and negative occurrences of atoms in a clause in linear time (in the size of the given clause):

Proposition 3.6 (i) *F is a positive subformula of F.*

(ii) *If $\neg G$ is a positive (resp., negative) subformula of F, then G is a negative (resp., positive) subformula of F.*

(iii) *If $G \vee H$ is a positive subformula of F, then G and H are both positive subformulas of F.*

(iv) *If $G \wedge H$ is a negative subformula of F, then G and H are both negative subformulas of F.*

(v) *If $G \rightarrow H$ is a positive subformula of F, then G is a negative subformula and H is a positive subformula of F.*

(vi) *If $G \rightarrow \perp$ is a negative subformula of F, then G is a positive subformula of F.*

(vii) *F is positive in a clause Σ if it is an element of Σ.*

3.3 General Ordered Resolution

Let $\succ$ be an ordering and S be a selection function. General ordered resolution $\mathsf{O}_S^{\succ}$ is given by this inference system:

General ordered resolution with selection:

$$\frac{\Sigma_1[A_1] \;\; \ldots \;\; \Sigma_n[A_n] \quad \Delta[A_1, \ldots, A_n]}{\Sigma_1[A_1/\perp], \ldots, \Sigma_n[A_n/\perp], \Delta[A_1|\top, \ldots, A_n|\top]}$$

where (i) either $A_1, \ldots, A_n$ is selected by S in Δ, or else $S(\Delta)$ is empty, $n = 1$, and A_1 is maximal in Δ, (ii) each atom A_i is maximal in Σ_i, and (iii) no clause Σ_i contains a selected atom.

Ordered self-resolution:

$$\frac{\Delta, \Gamma[A]}{\Delta, \Gamma[A|\perp], \Gamma[A/\top]}$$

where (i) the atom A is maximal in Σ, and (ii) the premise contains no selected atom.

The subscript and/or superscript in $\mathsf{O}_S^{\succ}$ will be omitted if the ordering $\succ$ or the selection function S are clear from the context, or are intentionally left unspecified. Specific settings of these parameters will be discussed later, see Sections 7 and 8.

The last premise, Δ, in an ordered resolution inference is called the *negative* premise, whereas the Σ_i are called the *positive* premises. In an ordered resolution inference either all selected atoms or, if there are no selected atoms, the maximal atom in the negative premise are resolved. Furthermore, positive premises and the premises of self-resolution inferences must contain no selected atoms at all. All occurrences of the resolved atoms in the positive premises are replaced (by $\perp$) whereas replacement in the negative premise (by $\top$) need not be exhaustive. At least one occurence of each of the A_i must be replaced, though. For self-resolution, replacement is restricted to a subclause Γ. Substitution of A by $\top$ is exhaustive, while replacement by $\perp$ may be partial. For instance, to facilitate the lifting of inference rules to clauses with variables, it will be useful if not all occurrences of an atom have to be substituted.

Lemma 3.7 *Let $\succ$ be a total admissible ordering and S be any selection function. Then the conclusion of any inference in $\mathsf{O}_S^{\succ}$ is smaller than the negative premise.*

The lemma would not hold if the conclusions of these inferences were written as a disjunction, instead of as a multiset.

The way the inferences have been formulated might suggest that all possible forms of partial replacement have to be considered as conclusions. We shall see in Section 6.2 that this nondeterminism is don't care. As soon as one conclusion has been derived the others become redundant. Don't-care non-deterministic is also the selection of that part of the premise of a self-resolution inference in which substitution takes place.

4 Derivations and Redundancy

Redundancy is a key concept in theorem proving, especially in saturation-based approaches. Experiments with theorem provers indicate that mechanisms for deleting redundant formulas and avoiding redundant inferences, such as tautology deletion, subsumption, and critical pair criteria, are indispensable for practical performance. We shall present an abstract notion of redundancy from which virtually all commonly used simplification techniques can be derived as special cases. Refutational completeness will be established with respect to this general notion of redundancy. Redundancy is a concept referring to a particular state in a theorem proving process. These states have to be modelled by adequate data structures to be introduced next.

4.1 Matrices and Sequents

A *matrix* is a (finite or infinite) multiset of finite multisets of formulas. More specifically, we distinguish between *positive matrices* that represent disjunctions of conjunctions; and *negative matrices* that represent conjunctions of disjunctions. Hence, a negative [positive] matrix is a finite or infinite multiset of [dual] clauses. Consequently, the negation $\neg M$ of a finite negative [positive] matrix will denote the disjunction [conjunction] of their negated [dual] clauses. We shall use the letters K, N and M, respectively, to denote matrices. A *sequent* is a pair of matrices, written $N \vdash M$ consisting of a negative matrix N and a positive matrix M. The matrix N is called the *antecedent* and the matrix M, the *succedent* of the sequent. A sequent $N \vdash M$ is true in an interpretation I if, whenever all clauses in N are true in I, then there exists a multiset (of formulas) Λ in M, such that all formulas of Λ are true in I. An empty multiset represents the constant $\top$ in the antecedent and the constant $\perp$ in the succedent.

Notions such as logical equivalence will be applied to matrices and sequents in the expected manner. For example, the two sequents $N \vdash M$ and $N, \neg M \vdash \perp$ are logically equivalent.

In a context where $\succ$ is a given ordering on formulas, we assume clauses [and finite matrices] to be ordered by its [twofold] multiset extension.

Refutational theorem provers only consider sequents with an empty succedent, whereas connection and tableau calculi are usually formulated in terms of sequents with an empty antecedent. These "one-sided" sequents can be identified with their negative and positive matrices, respectively. Since the main purpose of this paper is the investigation of refutational theorem proving based on resolution calculi, all matrices below will be of the negative type, unless indicated otherwise.

4.2 Theorem Proving Processes

We start out by developing the minimal prerequisites for a theory of refutational theorem proving with deduction and deletion. Deduction one may assume to be based on an inference system $\mathcal{J}$ for matrices. Deletion is more subtle in that derived formulas may be deleted afterwards only if it can be assured that they are not needed for the eventual proof. Deletion must therefore be based on a suitable redundancy criterion for formulas and inferences.

For defining a redundancy criterion, two mappings $\mathcal{R}_{\mathcal{F}}$ and $\mathcal{R}_{\mathcal{I}}$ are employed. $\mathcal{R}_{\mathcal{F}}$ and $\mathcal{R}_{\mathcal{I}}$ associate with each matrix N of clauses a set of clauses (that is, a matrix with no multiple occurrences of clauses) and a set of inferences, respectively, which are deemed to be redundant in the context N.

Definition 4.1 *Let $\mathcal{J}$ be an inference system. A pair $\mathcal{R} = (\mathcal{R}_{\mathcal{F}}, \mathcal{R}_{\mathcal{I}})$ of mappings from sets of clauses to sets clauses and inferences, respectively, is called a* redundancy criterion (for $\mathcal{J}$) *if, for all sets of clauses N and N'*

(R1) *if $N \subseteq N'$ then $\mathcal{R}_{\mathcal{F}}(N) \subseteq \mathcal{R}_{\mathcal{F}}(N')$ and $\mathcal{R}_{\mathcal{I}}(N) \subseteq \mathcal{R}_{\mathcal{I}}(N')$;*
(R2) *if $N' \subseteq \mathcal{R}_{\mathcal{F}}(N)$ then $\mathcal{R}_{\mathcal{F}}(N) \subseteq \mathcal{R}_{\mathcal{F}}(N \setminus N')$ and $\mathcal{R}_{\mathcal{I}}(N) \subseteq \mathcal{R}_{\mathcal{I}}(N \setminus N')$;*
(R3) *if $N \setminus \mathcal{R}_{\mathcal{F}}(N)$ is satisfiable then N is satisfiable; and*
(R4) *if Σ is the conclusion of an inference I in $\mathcal{J}$ then I is in $\mathcal{R}_{\mathcal{I}}(N \cup \{\Sigma\})$.*

The first condition expresses monotonicity of redundancy under the subset relation and, hence, in particular under the deduction of new clauses. The second condition requires that redundancy of a formula or of an inference be independent of clauses redundant in the same context. The third requirement states that the removal of redundant clauses preserves unsatisfiability. Finally, (R4) implies that upon adding the conclusion of an inference to a matrix the inference becomes redundant henceafter. We emphasize that $\mathcal{R}_{\mathcal{F}}(N)$ need not be a subset of N and that $\mathcal{R}_{\mathcal{I}}(N)$ will usually contain inferences whose premises are not in N. Inferences in $\mathcal{R}_{\mathcal{I}}(N)$ and clauses in $\mathcal{R}_{\mathcal{F}}(N)$ are said to be *redundant (with respect to N)*. Note that if $\mathcal{J}'$ is an inference system such that $\mathcal{J} \subseteq \mathcal{J}'$ then if $\mathcal{R}$ is a redundancy criterion for $\mathcal{J}$ it can also be considered a redundancy criterion for $\mathcal{J}'$ which classifies none of the inferences in $\mathcal{J}' \setminus \mathcal{J}$ in any context as redundant.

At an abstract level a saturation-based theorem prover can be described by a binary relation $\rhd$ on matrices, which we call a *transition* or *derivation relation*. More specifically, we consider derivation relations where each step $N \rhd N'$ consists of either adding a clause or deleting a redundant one.

Let $\mathcal{J}$ be an inference system and let $\mathcal{R}$ be a redundancy criterion for $\mathcal{J}$. If N is a matrix, we denote by $\mathcal{J}(N)$ the set of all inferences from N. For a set of inferences I, $\mathcal{C}(\mathsf{I})$ shall denote the set of all conclusions of inferences in I.

Deduction:

$$N \rhd N, \Sigma \qquad \text{if } \Sigma \in \mathcal{C}(\mathcal{J}(N))$$

Deletion:

$$N, \Sigma \rhd N \qquad \text{if } \Sigma \in N \cup \mathcal{R}(N)$$

These two rules describe the basic steps in the theorem proving process. Since deduction determines which clauses need to be generated (the "search space"), a more restrictive inference system is preferable; whereas a powerful notion of redundancy is desirable for deletion, which provides a means of decreasing the search space.

In the context of refutational theorem proving, the basic requirement is preservation of consistency. We call an inference system $\mathcal{J}$ *consistency-preserving* if for all matrices N, the matrix $N \cup \mathcal{C}(\mathcal{J}(N))$ is consistent, whenever N is consistent. From now on we assume that inference systems be consistency-preserving. Note that by the condition (R3) any redundancy criterion is "inconsistency-preserving" in that the deletion of redundant formulas preserves inconsistency.

A (finite or countably infinite) sequence $N_0 \rhd N_1 \rhd N_2 \rhd \cdots$ is called a *(theorem proving) derivation.* A derivation is said to be *based on* $\mathcal{J}$ and $\mathcal{R}$, with $\mathcal{R}$ a redundancy criterion for $\mathcal{J}$, if every step is either by deduction with $\mathcal{J}$ or by deletion according to $\mathcal{R}$. The set $N_\infty = \bigcup_i \bigcap_{j \geq i} N_j$ of all *persisting* clauses is called the *limit* of the derivation. By a *theorem prover* we mean a procedure that accepts as input a matrix N, and produces a derivation $N = N_0 \rhd N_1 \rhd N_2 \rhd \cdots$ from N based on some inference system $\mathcal{J}$ and redundancy criterion $\mathcal{R}$. The sets N_i represent the successive states in the theorem proving process; the set N_∞ its result (which, in the case of an infinite derivation, is only obtained in the limit).

Lemma 4.2 *Let $N_0 \rhd N_1 \rhd N_2 \rhd \cdots$ be a derivation based on a $\mathcal{J}$ and $\mathcal{R}$. Then $\mathcal{R}_\mathcal{F}(\bigcup_j N_j) \subseteq \mathcal{R}_\mathcal{F}(N_\infty)$ and $\mathcal{R}_\mathcal{I}(\bigcup_j N_j) \subseteq \mathcal{R}_\mathcal{I}(N_\infty)$. Moreover, N_∞ is satisfiable if and only if N_0 is satisfiable.*

Proof. Let $\mathcal{R}$ stand for either $\mathcal{R}_\mathcal{F}$ or $\mathcal{R}_\mathcal{I}$. First note that, by (R1), $(\bigcup_j N_j) \setminus N_\infty \subseteq \mathcal{R}_\mathcal{F}(\bigcup_j N_j)$ and, hence, $(\bigcup_j N_j) \setminus \mathcal{R}_\mathcal{F}(\bigcup_j N_j) \subseteq N_\infty$. Again using (R1) we infer that $\mathcal{R}((\bigcup_j N_j) \setminus \mathcal{R}_\mathcal{F}(\bigcup_j N_j)) \subseteq \mathcal{R}(N_\infty)$. Condition (R2) states that $\mathcal{R}(N) \subseteq \mathcal{R}(N \setminus \mathcal{R}_\mathcal{F}(N))$, for any multiset N. We conclude that $\mathcal{R}(\bigcup_j N_j) \subseteq \mathcal{R}(N_\infty)$.

For the second part note that N_0 is satisfiable if and only if $\bigcup_j N_j$ is satisfiable. The assertion follows from the condition (R3) and the first half of the Lemma. □

Refutationally complete theorem provers produce derivations in which a contradiction is derived eventually whenever N_0 is unsatisfiable. To show refutational completeness one must assure that sufficiently many inferences are computed. In that regard two concepts, saturation and fairness, are of importance.

Let $\mathcal{J}$ and $\mathcal{J}'$ be two inference systems such that $\mathcal{J} \subseteq \mathcal{J}'$, and let $\mathcal{R}$ be a redundancy criterion for $\mathcal{J}$. In employing two inference systems, our intention is to distinguish between don't care inferences (in $\mathcal{J}' \setminus \mathcal{J}$) that represent simplification steps intented to increase efficiency and those inferences (in $\mathcal{J}$) which are required for refutational completeness. We say that N is *saturated up to redundancy* (with respect to $\mathcal{J}$ and $\mathcal{R}$) if all inferences in $\mathcal{J}$ with non-redundant premises from N are redundant in N, i.e., $\mathcal{J}(N \setminus \mathcal{R}_\mathcal{F}(N)) \subseteq \mathcal{R}_\mathcal{I}(N)$. Saturation can be achieved by fair computation A derivation $N_0 \rhd N_1 \rhd N_2 \rhd \cdots$ based on $\mathcal{J}'$ and $\mathcal{R}$ is called *fair* with respect to $\mathcal{J} \subseteq \mathcal{J}'$ if every inference in $\mathcal{J}$ with non-redundant premises in N_∞ is redundant with respect to $\bigcup_j N_j$, i.e., $\mathcal{J}(N_\infty \setminus \mathcal{R}_\mathcal{F}(N_\infty)) \subseteq \mathcal{R}_\mathcal{I}(\bigcup_j N_j)$.

Fairness essentially requires that no inference in $\mathcal{J}$ from non-redundant persisting formulas be delayed indefinitely. A sufficient condition for fairness is expressed in the following lemma.

Lemma 4.3 *A derivation is fair with respect to $\mathcal{J}$ if the conclusion of every non-redundant inference in $\mathcal{J}$ from non-redundant formulas in N_∞ is an element of, or is redundant in, $\bigcup_j N_j$, i.e.,*

$$\mathcal{C}(\mathcal{J}(N_\infty \setminus \mathcal{R}_\mathcal{F}(N_\infty)) \setminus \mathcal{R}_\mathcal{I}(N_\infty \setminus \mathcal{R}_\mathcal{F}(N_\infty)))) \subseteq (\bigcup_j N_j \cup \mathcal{R}_\mathcal{F}(\bigcup_j N_j)).$$

Proof. If I is a redundant inference from non-redundant formulas in N_∞ then, by (R1), I is also redundant in $\bigcup_j N_j$. Let now I be a non-redundant inference from non-redundant formulas in N_∞ and let Σ be its conclusion. If Σ is in $\bigcup_j N_j$ then, by the condition (R4), I is redundant in $\bigcup_j N_j$. Otherwise, observe that I is in $\mathcal{R}_\mathcal{I}(\bigcup_j N_j \cup \{\Sigma\})$, and that $\mathcal{R}_\mathcal{I}(\bigcup_j N_j) = \mathcal{R}_\mathcal{I}(\bigcup_j N_j \cup \{\Sigma\})$ if Σ is redundant in $\bigcup_j N_j$. $\square$

In other words, a fair derivation can be constructed by exhaustively applying inferences to persisting formulas.

Lemma 4.4 *If a derivation is fair with respect to $\mathcal{J}$ and $\mathcal{R}$ then its limit is saturated up to redundancy with respect to $\mathcal{J}$ and $\mathcal{R}$.*

Proof. If a derivation is fair, then $\mathcal{J}(N_\infty \setminus \mathcal{R}_\mathcal{F}(N_\infty)) \subseteq \mathcal{R}_\mathcal{I}(\bigcup_j N_j)$. By Lemma 4.2, $\mathcal{R}_\mathcal{I}(\bigcup_j N_j) \subseteq \mathcal{R}_\mathcal{I}(N_\infty)$. Thus, $\mathcal{J}(N_\infty \setminus \mathcal{R}_\mathcal{F}(N_\infty)) \subseteq \mathcal{R}_\mathcal{I}(N_\infty)$, which means that N_∞ is saturated up to redundancy. $\square$

In summary we obtain the following scheme for obtaining refutationally complete theorem provers. (i) Define $\mathcal{J}$ and $\mathcal{R}$. (ii) Show that $\mathcal{J}$ is consistency-preserving and that $\mathcal{R}$ satisfies the conditions of a redundancy criterion. (iii) Show that $\mathcal{J}$ together with $\mathcal{R}$ is *refutationally complete*, that is, for all matrices N, whenever N saturated up to redundancy with respect to $\mathcal{J}$ and $\mathcal{R}$ and is inconsistent, then N contains a contradiction. (iv) Design additional inferences for simplification, that is, inferences which derive formulas in order to make other, "larger" formulas redundant. These additional inferences constitute $\mathcal{J}' \setminus \mathcal{J}$. (v) Design effective (possibly incomplete) methods for detecting redundancy of formulas and inferences. Finally, (vi) assure that theorem proving processes based on $\mathcal{J}'$ and $\mathcal{R}$ fairly enumerate the inferences in $\mathcal{J}$.

Theorem 4.5 *Let $\mathcal{J}$ be a consistency-preserving inference system, and let $\mathcal{R}$ be a redundancy criterion for $\mathcal{J}$ such that $\mathcal{J}$ is refutationally complete with respect to $\mathcal{R}$. Moreover let $\mathcal{J}' \supseteq \mathcal{J}$ be a sound extension of $\mathcal{J}$ and consider any theorem proving derivation $N_0 \rhd N_1 \rhd N_2 \rhd \cdots$ based on $\mathcal{J}'$ and $\mathcal{R}$ that is fair with respect to $\mathcal{J}$. Then N_0 is inconsistent if and only if N_∞ contains a contradiction.*

4.3 A Redundancy Criterion for General Resolution

We shall define a redundancy criterion $\mathsf{R}^{\succ}$, which we shall refer to as the *standard redundancy criterion* for general ordered resolution. The criterion will be based on the given clause ordering $\succ$, but will be independent of selection. The criterion is a slight optimization of the one proposed by Bachmair & Ganzinger (1990), (also see Bachmair & Ganzinger 1994*a*). For the purposes of this definition, the ordering $\succ$ need not be an admissible ordering. It will be sufficient to require that $\succ$ be a well-founded partial ordering on clauses.

Definition 4.6 *A (general) clause Σ is called* redundant in N *(with respect to $\succ$) if there exist clauses $\Sigma_1, \ldots, \Sigma_k$ in N such that $\Sigma_1, \ldots, \Sigma_k \models \Sigma$ and $\Sigma \succ \Sigma_1, \ldots, \Sigma_k$.*

In other words, a clause is redundant in N if it is a logical consequence of smaller clauses in N. For example, tautologies $\Sigma, \top$ or $\Sigma, A, \neg A$ are redundant in any context N. Note that Σ need not be an element of N in order to be redundant in N. Let us denote by $N_{\Sigma}^{\succ}$ the set of all clauses Δ in N, such that $\Sigma \succ \Delta$. Σ is redundant, if and only if $N_{\Sigma}^{\succ} \models \Sigma$.

By $\mathcal{R}_{\mathcal{F}}^{\succ}(N)$ we denote the set of all clauses that are redundant in N with respect to $\succ$ by the standard criterion.

Lemma 4.7 *If $N \subseteq N'$, then $\mathcal{R}_{\mathcal{F}}^{\succ}(N) \subseteq \mathcal{R}_{\mathcal{F}}^{\succ}(N')$. Furthermore, if a clause Σ is redundant in N, then there exist non-redundant clauses $\Sigma_1, \ldots, \Sigma_k$ in N, such that $\Sigma_1, \ldots, \Sigma_k \models \Sigma$ is valid and $\Sigma \succ \Sigma_1, \ldots, \Sigma_k$. Consequently, $\mathcal{R}_{\mathcal{F}}^{\succ}(N) \subseteq \mathcal{R}_{\mathcal{F}}^{\succ}(N \setminus \mathcal{R}_{\mathcal{F}}^{\succ}(N))$, for all matrices N.*

Sketch of proof. Part (1) follows immediately from the definiton of redundancy. For part (2), let us suppose Σ is redundant in N. Let $N' = \Sigma_1, \ldots, \Sigma_k$ be a minimal submultiset of N (with respect to the multiset ordering $\succ_{mul}$), such that $\Sigma_1, \ldots, \Sigma_k \models \Sigma$ and $\Sigma \succ \Sigma_j$, for all j. The clauses Σ_j are all non-redundant. □

Redundancy of resolution inferences is defined as follows:

Definition 4.8 *An ordered resolution inference with positive premises $\Sigma_1, \ldots, \Sigma_n$, negative premise Δ, and conclusion Σ' is called* redundant *in N (with respect to $\succ$) if either (i) Σ' is in N or else (ii) there exist clauses $\Delta_1, \ldots, \Delta_k$ in N such that $\Delta \succ \Delta_1, \ldots \Delta_k$, and*

$$\Delta_1, \ldots, \Delta_k, \Sigma_1, \ldots, \Sigma_n \models \Sigma'.$$

An ordered self-resolution inference with premise Σ, resolved atom A and conclusion Σ' is called redundant *in N (with respect to $\succ$) if either (i) Σ' is in N or else (ii) there exist clauses $\Delta_1, \ldots, \Delta_k$ in N such that $\Delta \succ \Delta_1, \ldots \Delta_k$, and*

$$\Delta_1, \ldots, \Delta_k, \neg\Sigma[A/\top], \neg A \models \Sigma'.$$

In both cases the inference is redundant if its conclusion is entailed by clauses $\Delta_1, \dots, \Delta_k$ in N which are smaller that the (negative) premise, where certain subformulas of the inference may be used as additional assumptions. For ordered resolution one may additionally assume the positive premises. For self-resolution one has the negation of the resolved atom A and the negation of the premise, with A replaced by $\perp$, available. Note that if $\succ$ is total, by the Lemma 3.7, the first alternative (i) in the definition of redundancy is subsumed by the second alternative (ii).

Clearly, an inference is redundant whenever the resolvent Σ' is redundant, but the converse is not true in general for several reasons. The Δ_i have to be smaller than the [negative] premise but may generally be larger than Σ' as the ordered inferences are monotone. For ordered resolution validity of the positive premises may be assumed for a redundancy proof. They might not be in N or they might be larger than the negative premise. Redundancy of the inference neither implies redundancy of any of its premises. For example, let N consist of the clauses $(A \to B \wedge C)$ and $(A \wedge C \to E)$, and assume that B is maximal among the atoms. Neither $B \to E$ nor $(A \to B \wedge C)$ are redundant in N. However, the inference

$$\frac{(A \to B \wedge C) \quad (B \to E)}{(A \to \perp \wedge C)\,,\,(\top \to E)}\;,$$

the conclusion of which is equivalent to $A \to E$, is redundant as

$$(A \wedge C \to E),\ (A \to B \wedge C) \models A \to E.$$

Note that $A \wedge C \to E$ will be larger than the conclusion in most clause orderings and $A \to B \wedge C$ might even be the maximal premise of the inference. The following observation will often be implicitly applied in proofs of redundancy:

Proposition 4.9 *An inference by ordered resolution with positive premises $\Sigma_1, \dots, \Sigma_n$, negative premise Δ, resolved atoms $A_1, \dots, A_n$, and conclusion Σ' is redundant in N if there exist clauses $\Delta_1, \dots, \Delta_k$ in N such that $\Delta \succ \Delta_1, \dots, \Delta_k$ and*

$$\Delta_1, \dots, \Delta_k, \Sigma_1, \dots, \Sigma_n, A_1, \dots, A_n \models \Sigma'.$$

Proof. We have to show that under the given assumptions the implication

$$\Delta_1, \dots, \Delta_k, \Sigma_1, \dots, \Sigma_n \models \Sigma'$$

is valid. In fact, suppose the Δ_i and the Σ_i are true in an interpretation I but Σ' is false in I. As Σ' takes the form $\Sigma_1[A_1/\perp], \dots, \Sigma_n[A_n/\perp], \Delta'$, the A_i must be true in I. Therefore Σ' is true in I, which is a contradiction. □

For ordered self-resolution, redundancy of an inference again does not imply the redundancy of neither its conclusion nor its premise. Suppose $\Sigma = (\neg A \wedge D,\, A \wedge B,\, A \wedge C)$ is split into parts $\Delta = (\neg A \wedge D,\, A \wedge B)$ and $\Gamma = (A \wedge C)$. Then the conclusion of self-resolving A in the Γ-part of Σ produces a clause equivalent to $\Sigma' = (\neg A \wedge D,\, A \wedge B,\, C)$. If N, in addition to Σ, just contains the clause (A, B, D), and assuming that A is the maximal atom and $\Sigma \succ (A, B, D)$, then neither Σ nor Σ' are redundant. On the other hand, (A, B, D), $\neg B \wedge \neg C$, $\neg A \models \Sigma'$, demonstrating the redundancy of the inference. In this example the two additional assumptions $\neg A$ and $\neg B \wedge \neg C$ (the latter is equivalent to $\neg\Sigma[A/\top]$) that one may exploit in a redundancy proof of an ordered self-resolution inference are essential. Dropping any one of them would no longer allow to deduce Σ' from (A, B, D).

By $\mathcal{R}^{\succ}_{\mathcal{I}}(N)$ we denote the set of resolution inferences that are redundant in N with respect to $\succ$ by the standard criterion. As an immediate consequence of the Lemma 4.7 we obtain:

Lemma 4.10 *If $N \subseteq N'$, then $\mathcal{R}^{\succ}_{\mathcal{I}}(N) \subseteq \mathcal{R}^{\succ}_{\mathcal{I}}(N')$. Moreover, $\mathcal{R}^{\succ}_{\mathcal{I}}(N) \subseteq \mathcal{R}^{\succ}_{\mathcal{I}}(N \setminus \mathcal{R}^{\succ}_{\mathcal{F}}(N))$, for all matrices N.*

Theorem 4.11 *$\mathcal{R}^{\succ}$ is a redundancy criterion for $\mathsf{O}^{\succ}$.*

Proof. Properties (R1) and (R2) follow from the Lemmas 4.7 and 4.10. Moreover, the redundancy criterion $\mathcal{R}^{\succ}$ is obviously sound in the sense of (R3). (R4) follows directly from the definition. □

5 Refutational Completeness

For demonstrating the refutational completeness of general ordered resolution $\mathsf{O}^{\succ}$ with respect to the standard redundancy criterion $\mathcal{R}^{\succ}$ by applying the Theorem 4.5 we need to show that saturated sets of clauses that do not contain a contradiction are satisfiable. We use the same model construction technique as for the proof of Theorem 3.4 above, but apply it to general clauses. Let N be a negative matrix and $\succ$ be a *total* admissible ordering on clauses. By induction over $\succ$ we define a Herbrand interpretation I_Σ and a set ε_Σ, for each clause Σ in N as follows:

Definition 5.1 *Let I_Σ be the set $\bigcup_{\Sigma \succ \Delta} \varepsilon_\Delta$. Furthermore, if A is the maximal atomic formula in Σ, then $\varepsilon_\Sigma = \{A\}$ if (i) $A \notin I_\Sigma$, and (ii) Σ is false in I_Σ, but Σ is true in $I_\Sigma \cup \{A\}$. Otherwise, ε_Σ is the empty set.*

We say that Σ *produces* A, and call Σ *productive*, if $\varepsilon_\Sigma = \{A\}$. By I_N we denote the Herbrand interpretation $\bigcup_{\Sigma \in N} \varepsilon_\Sigma$.

Lemma 5.2 *Let N be a set of ground clauses, $\succ$ a total admissible ordering, and I the interpretation constructed from N. If N is saturated up to redundancy and contains no contradiction, then for all atoms A and all clauses Σ in N with maximal atom A:*

1. *If Σ produces A then Σ is non-redundant, contains no selected atoms, and for all clauses Σ' in N, where $\Sigma \succ \Sigma'$, Σ' is true in I_Σ.*
2. *Σ is true in I.*

Proof. The proof is by induction. Let A be an atomic formula. We assume that properties 1. and 2. hold for clauses with a maximal atom B such that $A \succ B$.

1. Suppose Σ produces A and let Σ' be a clause in N with $\Sigma \succeq \Sigma'$. We will show that either (i) Σ' is true in I_Σ or else (ii) $\Sigma = \Sigma'$, Σ' is non-redundant, and Σ' does not contain a selected atom. Note that A is not in I_Σ. The proof of (i) is by induction over Σ'. We assume that (i) is true for all clauses Γ in N with $\Sigma' \succ \Gamma$.

(1) If the maximal atom in Σ' is smaller than A then, by part 2. of the main induction hypothesis, Σ' is true in I. Since the interpretations I and I_Σ assign the same truth value to all atoms strictly smaller than A, Σ' is then also true in I_Σ.

(2) Suppose now that Σ' contains A.

(2.1) If Σ' is redundant in N then there exist clauses Γ_i in N such that $\Sigma' \succ \Gamma_i$ which logically imply Σ'. By induction hypothesis these clauses are true in I_Σ, hence so is Σ'.

(2.2) Suppose now that Σ' is not redundant. We distinguish whether or not Σ' contains a selected atom.

(2.2.1) Suppose Σ' contains no selected atom. If $\Sigma = \Sigma'$ then (ii) follows. Otherwise, let $\Sigma \succ \Sigma'$. As Σ produces A, the clause Σ', having A as maximal atom, cannot be productive, and, therefore, I_Σ and $I_{\Sigma'}$ are identical. Suppose, for the purpose of deriving a contradiction, that Σ' is false in I_Σ. If $\Sigma'[A/\top]$ were true in I_Σ, the clause Σ' would be productive. Hence $\neg\Sigma'[A/\top]$ is true in I_Σ. Consider any self-resolution inference with premise $\Sigma' = \Delta, \Gamma[A]$ and resolvent $\Sigma'' = (\Delta, \Gamma[A|\bot], \Gamma[A/\top])$. Since N is saturated up to redundancy and Σ' is non-redundant, the inference must be redundant in N. Hence there exist clauses Γ_i in N such that $\Sigma' \succ \Gamma_i$ and Σ'' is true in any interpretation satisfying the Γ_i, $\neg\Sigma'[A/\top]$, and $\neg A$. By the induction hypothesis, the Γ_i are true in I_Σ, as are $\neg A$ and $\neg\Sigma'[A/\top]$. Consequently, Σ'' is true in I_Σ. Note that $\Delta, \Gamma[A|\bot]$ and Σ' have the same thruth value in I_Σ. As Σ'' is true in I_Σ, the clause $\Gamma[A/\top]$ must be true in I_Σ. But then Σ' would be true in $I_{\Sigma'} \cup \{A\}$ and produce A, which is a contradiction. Hence Σ' is true in I_Σ.

(2.2.2) Now assume that $S(\Sigma')$ is nonempty. Suppose that Σ' is false in I_Σ. By the defining properties of selection functions, we may find a sequence $A_1, \ldots, A_n$ of atoms in $S(\Sigma')$ which are all true in I_Σ, hence $A \succ A_i$. Let Δ_i be the clause that produces A_i. Since any Δ_i contains only atoms smaller than A, we may use the (main) induction hypothesis to infer that the Δ_i are non-redundant and contain no selected atoms. From the premises

$\Delta_1, \ldots, \Delta_n$ and Σ we obtain a resolvent

$$\Sigma'' = \Delta_1[A_1/\bot], \ldots, \Delta_n[A_n/\bot], \Sigma'[A_1|\top, \ldots, A_n|\top].$$

Since all premises are non-redundant, and N is saturated up to redundancy, the resolvent Σ'' is a logical consequence of the positive premises Δ_i and those clauses in N that are smaller than Σ'. By the induction hypothesis all such clauses are true in I_Σ. Since Δ_i produces A_i, the clause $\Delta_i[A_i/\bot]$ is false in I_{Δ_i} and, hence, in I_Σ. Therefore, $\Sigma'[A_1|\top, \ldots, A_n|\top]$ must be true in I_Σ. As $A_i \in I_\Sigma$, for all i, the truth value of the latter clause and the truth value of Σ' coincide in I_Σ. In short, Σ' is true in I_Σ.

2. We prove this part by induction on Σ. If Σ is redundant the assertion follows immediately from the induction hypothesis. Also, if Σ is productive it is true in I. Hence we may assume that Σ is non-redundant and non-productive.

(1) Suppose that $S(\Sigma)$ is nonempty and that Σ is false in I. Then there exists a sequence $A_1, \ldots, A_n$ of atoms in $S(\Sigma)$ which are all true in I; let Δ_i be the clauses that produce the A_i. By part (1) any of the Δ_i is non-redundant and contains no selected atoms. From premises $\Delta_1, \ldots, \Delta_n$ and Σ we obtain a resolvent

$$\Sigma' = \Delta_1[A_1/\bot], \ldots, \Delta_n[A_n/\bot], \Sigma[A_1|\top, \ldots, A_n|\top].$$

Since all premises are non-redundant, and N is saturated up to redundancy, Σ' is a logical consequence of the Δ_i and those clauses in N that are smaller than Σ. The Δ_i and A_i are true in I. By the induction hypothesis, clauses smaller than Σ are true in I. Altogether, Σ' is true in I. Since Δ_i produces A_i, $\Delta_i[A_i/\bot]$ must be false in I_{Δ_i} and, hence, in I. Therefore, $\Sigma[A_1|\top, \ldots, A_n|\top]$, and hence Σ, must be true in I.

(2) Assume now that Σ is non-redundant and contains no selected atom. We distinguish two cases:

(2.1) Suppose that A is not in I. Consider any self-resolution inference with premise $\Sigma = \Delta, \Gamma[A]$ and resolvent $\Sigma' = (\Delta, \Gamma[A|\bot], \Gamma[A/\top])$. Since N is saturated up to redundancy and Σ' is non-redundant, the inference must be redundant in N. Hence there exist clauses Γ_i in N such that $\Sigma \succ \Gamma_i$ which, together with $\neg A$ and $\neg\Sigma[A/\top]$, logically imply Σ'. If $\Sigma[A/\top]$ is true in I then Σ is also true in I. For, otherwise, Σ would have to produce A which contradicts the assumption that A is not in I. Let us now assume that $\neg\Sigma[A/\top]$ is true in I. By the induction hypothesis, the Γ_i are true in I. We conclude that the conclusion Σ' of the self-resolution inference is true in I. Suppose Σ were false in I. As we have assumed that A is not in I, the clauses Σ and $(\Delta, \Gamma[A|\bot])$ have the same truth value in I hence $\Gamma[A/\top]$ must be true in I, and, therefore, Σ true in $I \cup \{A\}$. Then Σ produces A, which is a contradiction. Hence Σ is true in I.

(2.2) Finally, suppose that A is in I, and let Δ be the clause that produces A. Consider the non-clausal resolvent

$$\Sigma' = \Delta[A/\bot], \Sigma[A|\top],$$

of Δ and Σ. Since N is saturated up to redundancy, the resolvent must be a logical consequence of Δ, of A and the clauses in N that are smaller than Σ. Since all these clauses are true in I, we may infer that Σ' is also true in I. Consequently, Σ is true in I. $\square$

We call an ordering $\succ$ on formulas *completable* if it can be extended to a total admissible ordering. An ordering on multisets is *completable* if it is the multiset extension of a completable ordering on formulas.

Redundancy of a clause or inference is preserved under extension of the underlying ordering.

Lemma 5.3 *If the ordering $\succ'$ extends $\succ$, then $\mathcal{R}^{\succ}_{\mathcal{F}}(N) \subseteq \mathcal{R}^{\succ'}_{\mathcal{F}}(N)$ and $\mathcal{R}^{\succ}_{\mathcal{I}}(N) \subseteq \mathcal{R}^{\succ'}_{\mathcal{I}}(N)$. Consequently, if N is saturated up to redundancy under $\mathsf{O}^{\succ}_{S}$, then it is also saturated up to redundancy under $\mathsf{O}^{\succ'}_{S}$.*

Theorem 5.4 (Refutational completeness) *Let $\succ$ be a completable ordering and S be a selection function. If N is saturated up to redundancy under $\mathsf{O}^{\succ}_{S}$, then N is unsatisfiable if and only if it contains a contradiction.*

Proof. If N contains a contradiction, then it is unsatisfiable. Suppose N contains no contradiction and let $\succ'$ be a total admissible ordering that extends $\succ$. By Lemma 5.3, if N is saturated up to redundancy under $\mathsf{O}^{\succ}_{S}$, then it is also saturated up to redundancy under $\mathsf{O}^{\succ'}_{S}$. We may use Lemma 5.2, to infer that N has a model. $\square$

6 Applications of Standard Redundancy

The purpose of this section is to indicate how wide a range of resolution-based theorem proving strategies is provided by the general completeness results in the Theorems 4.5 and 5.4. We shall begin with describing further restrictions to the inference system that are furnished by the standard redundancy criterion.

6.1 Polarity-Based Restrictions

Inferences which yield tautologies are trivially redundant. An analysis of the polarity of the resolved atoms is helpful in this regard.

Proposition 6.1 *An inference by ordered resolution is redundant if the negative premise contains a positive occurrence of any of the resolved atoms or if any of the positive premises Σ_i contains a negative occurrence of A_i or a positive occurrence of any of the atoms A_j with $A_j \neq A_i$.*

Proof. Let $\Sigma_i[A_i]$, $1 \leq i \leq n$, be the positive premises, and let Δ be the negative premise of the inference, resolving $A_1, \ldots, A_n$. Then the resolvent is of the form $\Sigma' = \Sigma_1[A_1/\bot], \ldots, \Sigma_n[A_n/\bot], \Delta[A_1|\top, \ldots, A_n|\top]$. If, say, A_1 occurs positively in Δ then $\Delta[A_1|\top, \ldots, A_n|\top]$ is true in any interpretation in which A_1 is true. Thus, A_1 entails Σ', and by the Proposition 4.9 the inference is redundant. If Σ_i contains A_i negatively, then $\Sigma_i[A_i/\bot]$ is a tautology, and so is Σ'. If, say, A_2 occurs positively in, say, Σ_1, then $\Sigma_1[A_2/\top]$ is a tautology, as is $\Sigma_1[A_1/\bot, A_2/\top]$, provided $A_1 \neq A_2$. Therefore, A_2 entails $\Sigma_1[A_1/\bot]$ and hence the conclusion Σ'. This proves the redundancy of the inference. $\square$

Proposition 6.2 *An inference by self-resolution is redundant if the resolved atom occurs positively or negatively in the premise.*

Proof. Let $\Sigma = (\Delta, \Gamma[A])$ be the premise, A the resolved atom, and let $\Sigma' = (\Delta, \Gamma[A|\bot], \Gamma[A/\top])$ denote the conclusion of the inference. If A occurs positively in Σ, the clause $\neg\Sigma[A/\top]$ is false in any interpretation, therefore entails Σ', and the inference is redundant. If A occurs negatively in Σ, the clause $\Sigma[A/\bot]$ is true in any interpretation. Consequently $(\Delta, \Gamma[A|\bot])$ is true in any interpretation in which A is false. Therefore Σ' is entailed by $\neg A$, hence the inference is redundant. $\square$

Propositions 6.1 and 6.2 give rise to additional polarity constraints that we may safely attach to ordered [self-] resolution.

Let us briefly remark that Manna & Waldinger (1980) and Murray (1982) proposed a different restriction for general resolution, where the resolved atom A is required to occur positively in the positive premise and negatively in the negative premise. This requires a different notion of polarity, according to which each subformula is positive or negative (or both). For instance, A and B are considered to be positive in $A \wedge B$. These polarity constraints are not compatible with ordering constraints or selection. The combination yields an incomplete calculus. For example, take the formulas $A \wedge B$ and $\neg B$ and suppose $A \succ B$ and A and B are considered positive in $A \wedge B$. The inference

$$\frac{A \wedge B \quad \neg B}{(A \wedge \bot), \neg\top}$$

satisfies the polarity constraint: B is positive in the first and negative in the second premise; but not the ordering constraint: B is not maximal in the first premise. (The resolvent, by the way, is a contradictory formula, but not a contradiction in our sense.) On the other hand, the inference

$$\frac{A \wedge B \quad A \wedge B}{(\bot \wedge B), (\top \wedge B)}$$

satisfies the ordering constraint, but not the polarity constraint, as A occurs only positively in both premises. The resolvent of this inference is equivalent to B; another resolution step with $\neg B$ yields a contradiction. There is no resolution inference from $A \wedge B$ and $\neg B$ that satisfies both polarity and ordering constraints. In other words, the simultaneous application of both kinds of constraints renders non-clausal resolution incomplete.

6.2 Replacement Strategies and Implicit Factoring

Resolution inferences are essentially case analyses with respect to certain atoms. The atom on which a case split is performed is substituted by its possible truth values $\top$ or $\bot$. We have formulated the general inferences in a way such that to a large extent the particular choice of which occurrences are to be substituted has been left open. The following propositions show that the actual choice can be made don't-care non-deterministically.

Proposition 6.3 *Consider any two ordered resolution inferences from the same premises resolving the same atoms. If one inference is redundant in N so is the other.*

Proof. Let Σ' and Σ'' be two resolvents derived by ordered resolution from positive premises $\Sigma_1, \ldots, \Sigma_n$ and a negative premise Δ. Σ' and Σ'' can differ only in which occurrences of the resolved atoms have been replaced (by $\top$) in the negative premise. Therefore, if $A_1, \ldots, A_n$ are the resolved atoms, Σ and Σ' are equivalent in any model of $A_1, \ldots, A_n$. Suppose that the inference which derives one of the resolvents, Σ' say, is redundant in N. That means $\Delta_1, \ldots, \Delta_k, \Sigma_1, \ldots, \Sigma_n, A_1, \ldots, A_n \models \Sigma'$ for certain clauses Δ_i in N which are smaller than Δ. Then $\Delta_1, \ldots, \Delta_k, \Sigma_1, \ldots, \Sigma_n, A_1, \ldots, A_n \models \Sigma''$, and by the extended redundancy criterion (cf. Proposition 4.9) redundancy of the other inference follows. $\Box$

In a similar way we obtain the corresponding result for self-resolution.

Proposition 6.4 *Consider any two ordered self-resolution inferences from the same premise resolving the same atom. If one inference is redundant in N so is the other.*

Proof. Two self-resolution inferences of the indicated kind may differ in the subfomula Δ and Δ', respectively, in which resolution of the atom A takes place, and in the selection of positions in Δ and Δ', respectively, at which A gets replaced by $\bot$. Hence $\Sigma' = (\Gamma, \Delta, \Delta'[A|\bot], \Delta'[A/\top])$ and $\Sigma'' = (\Gamma, \Delta', \Delta[A|\bot], \Delta[A/\top])$ are the two resolvents from the premise $\Sigma = (\Gamma, \Delta, \Delta')$. One observes that interpretations in which both $\neg A$ and $\neg\Sigma[A/\top]$ are true assign the same truth value to both Σ' and Σ''. Consequently, if one of them is implied by $\neg A$, $\neg\Sigma[A/\top]$, and certain clauses in N, so is the other. $\Box$

In the propositional case one usually substitutes many, if not all, occurrences of the atom that is resolved upon. For first-order clauses the simultaneous replacement of atoms requires *implicit or explcit factoring*. Different ground instances of a clause may have differently many occurrences of the maximal atom. As an example consider the clause $\neg p(x, g(y)) \vee \neg p(g(y), x) \vee p(x, y)$. Any ground instance in which $x = g(y)$ will have two occurrences of the maximal atom. In any other ground instance the maximal atom occurs just once. If one substitutes too few occurrences of the resolved atom then one forgets in too many places the information about the case analysis that the resolution inference performs. If one wants to substitute more occurences at the same time, one either has to implicitly factor by unifying different non-ground atoms in a clause before they are resolved or else one has to explicitly add a "factor" $\Sigma[A/A\sigma, B/B\sigma]$ of the clause Σ for maximal atoms A and B that are unifiable by σ. The disadvantage of the latter is that after factorization a certain set of ground clauses is represented twice by different non-ground clauses, the original clause, and its factor. This leads to a duplication of computation. For the positive premise there is no choice however. All occurrences of the resolved atom need to be replaced, hence a certain amount of implicit or explicit factoring is required.

We will speak of *simple resolution* if substitution, where it may be partial, is confined to a single formula in a clause, but done exhaustively there.

Simple ordered resolution with selection:

$$\frac{\Sigma_1[A_1] \quad \ldots \quad \Sigma_n[A_n] \quad \Delta, F[A_1, \ldots, A_n]}{\Sigma_1[A_1/\bot], \ldots, \Sigma_n[A_n/\bot], \Delta, F[A_1/\top, \ldots, A_n/\top]}$$

where F is a formula such that (i) either $A_1, \ldots, A_n$ is selected by S in (Δ, F), or else $S(\Delta, F)$ is empty, $n = 1$, and A_1 is maximal in (Δ, F), (ii) each atom A_i is maximal in Σ_i, and (iii) no clause Σ_i contains a selected atom.

Simple ordered self-resolution:

$$\frac{\Delta, F[A]}{\Delta, F[A/\bot], F[A/\top]}$$

where (i) the atom A is maximal in Σ, and (ii) Δ, F contains no selected atom.

The refutational completeness of simple resolution follows from the above propositions together with the Theorem 5.4. The propositions also indicate that the choice of the particular formula F in the inferences is don't-care.

We end this section by showing that resolving only a subsequence of any selected sequence of atoms in a negative premise is sufficient.

Proposition 6.5 *Let*

$$\frac{\Sigma_1[A_1] \quad \ldots \quad \Sigma_n[A_n] \quad \Delta[A_1, \ldots, A_n]}{\Sigma_1[A_1/\bot], \ldots, \Sigma_n[A_n/\bot], \Delta[A_1|\top, \ldots, A_n|\top]}$$

be a general ordered resolution inference in which $n > 1$ and the $A_1, \ldots, A_n$ are selected in the negative premise. Let $\{i_1, \ldots, i_k\}$ be any non-empty subset of the indexes $1 \leq i \leq n$. The inference is redundant in N, whenever the "partial conclusion"

$$\Sigma = (\Sigma_{i_1}[A_{i_1}/\bot], \ldots, \Sigma_{i_k}[A_{i_k}/\bot], \Delta[A_{i_1}|\top, \ldots, A_{i_k}|\top])$$

is implied by the Σ_j and clauses in N smaller than Δ.

Proof. Let Γ_i, $1 \leq i \leq m$, be clauses in N such that

$$\Gamma_1, \ldots, \Gamma_m, \Sigma_1, \ldots, \Sigma_n \models \Sigma$$

Then the clauses

$$\Gamma_1, \ldots, \Gamma_m, A_1, \ldots, A_n, \Sigma_1, \ldots, \Sigma_n$$

imply the conclusion of the given inference, yielding the redundancy criterion of Proposition 4.9 □

A resolution inference in which simultaneously atoms $A_1, \ldots, A_n$ are resolved can be implemented by any sequence of inferences in which at each step only some of the atoms are resolved (using the corresponding subset of the positive premises).[2] The proposition says that if one such partial resolvent is generated or otherwise shown redundant, the original inference is redundant, too. In short, whenever a sequence of atoms is selected in a clause we may don't-care non-deterministically resolve a subset of it. The partial inference can be equipped with the same redundancy criterion as we have defined it for the resolution inferences proper.

6.3 Simplification

An important application of redundancy is in the use of logical equivalences for simplification of clauses. For instance, suppose $N, \Sigma' \models \Sigma$ and $N, \Sigma \models \Sigma'$ and $\Sigma \succ \Sigma'$. Then there is a two-step derivation,

$$N, \Sigma \rhd N, \Sigma, \Sigma' \rhd N, \Sigma'$$

where the first step is by deduction, as Σ' is a logical consequence of N, Σ; and the second by deletion, as Σ is rendered redundant by Σ'. We thus obtain a derived inference rule:

[2] These partial inferences are sound but do not satisfy the restriction about selection.

Simplification:

$$N, \Sigma \rhd N, \Sigma'$$

if $N, \Sigma' \models \Sigma$ and $N, \Sigma \models \Sigma'$ and $\Sigma \succ \Sigma'$

For example,

$$N, (\Sigma, \bot) \rhd N, \Sigma$$

is a simplification step.

A more interesting case of simplification is the use of object-level equivalences $F \leftrightarrow G$ for rewriting. More specifically, we get

$$N, (F \leftrightarrow G), \Sigma[F] \quad \rhd \quad N, (F \leftrightarrow G), \Sigma[G] \qquad \text{if } \Sigma[F] \succ \Sigma[G]$$

Equivalences occur in many problem domains and very often simplification is the natural way of dealing with them. For example, an equivalence $X \subseteq (Y \cap Z) \leftrightarrow [(X \subseteq Y) \wedge (X \subseteq Z)]$ can be used to replace any occurrence of $X \subseteq (Y \cap Z)$ by a conjunction of simpler "subset relations" $X \subseteq Y$ and $X \subseteq Z$. We believe that simplification in our sense is also the right framework for an analysis of the question of "demodulation across argument and literal boundaries," a research problem posed by Wos (1988).

Metalevel equivalences suitable for simplification can be conveniently described by rewrite systems. For example, by P we denote the set of the following rewrite rules for elimination of $\bot$ and $\top$ from conjunctions, disjunctions and negations:

$$\begin{array}{rcl@{\qquad}rcl}
\alpha \wedge \bot & \Rightarrow & \bot & \bot \wedge \alpha & \Rightarrow & \bot \\
\alpha \wedge \top & \Rightarrow & \alpha & \top \wedge \alpha & \Rightarrow & \alpha \\
\alpha \vee \bot & \Rightarrow & \alpha & \bot \vee \alpha & \Rightarrow & \alpha \\
\alpha \vee \top & \Rightarrow & \top & \top \vee \alpha & \Rightarrow & \top \\
\neg\bot & \Rightarrow & \top & \neg\top & \Rightarrow & \bot
\end{array}$$

If $F \Rightarrow F'$ is a ground instance of a rule in P, then $F \equiv F'$. Furthermore, the rewrite system is contained in any simplification ordering (including lexicographic path orderings). Consequently,

$$N, \Sigma \rhd N, \Sigma' \qquad \text{if } \Sigma \Rightarrow^{+}_{\mathsf{P}} \Sigma'$$

is a simplification step (for any simplification ordering).

Similar rules for the elimination of $\top$ and $\bot$ can be designed for other connectives, e.g.,

$$\begin{array}{rcl@{\qquad}rcl}
\alpha \rightarrow \bot & \Rightarrow & \neg\alpha & \bot \rightarrow \alpha & \Rightarrow & \top \\
\alpha \rightarrow \top & \Rightarrow & \top & \top \rightarrow \alpha & \Rightarrow & \alpha \\
\alpha \leftrightarrow \bot & \Rightarrow & \neg\alpha & \bot \leftrightarrow \alpha & \Rightarrow & \neg\alpha \\
\alpha \leftrightarrow \top & \Rightarrow & \alpha & \top \leftrightarrow \alpha & \Rightarrow & \alpha
\end{array}$$

cover implication and equivalence. The simplification rules for eliminating $\top$ and $\bot$ are often directly built into specialized variants of inferences for specific classes (normalforms) of formulas as discussed below.

6.4 Normal Forms

Rewrite systems also provide a convenient way of describing various normal forms. Let us briefly discuss negation, conjunctive, and disjunctive normal form. First note that all connectives can be expressed in terms of disjunction, conjunction and negation, as expressed by the rules,

$$\begin{array}{rcl} \alpha \leftrightarrow \beta & \Rightarrow & (\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha) \\ \alpha \rightarrow \beta & \Rightarrow & \neg\alpha \vee \beta \end{array}$$

for the case of implication and equivalence. Termination of these rules can be proved by a lexicographic path ordering (based on a precedence) in which the symbols to be eliminated ($\leftrightarrow$ and $\rightarrow$ in this case) have higher precedence than the other connectives (here $\wedge$, $\vee$ and $\neg$).

We may then push negation inside and eliminate double negations:

$$\begin{array}{rcl} \neg(\alpha \vee \beta) & \Rightarrow & \neg\alpha \wedge \neg\beta \\ \neg(\alpha \wedge \beta) & \Rightarrow & \neg\alpha \vee \neg\beta \\ \neg\neg\alpha & \Rightarrow & \alpha \end{array}$$

Termination of these rules requires a precedence in which $\neg \succ \vee$ and $\neg \succ \wedge$. The normal forms defined by these rules are also called *negation normal forms.*

From negation normal form we get to *conjunctive normal form* by applying distributivity rules:

$$\begin{array}{rcl} (\alpha \wedge \beta) \vee \gamma & \Rightarrow & (\alpha \vee \gamma) \wedge (\beta \vee \gamma) \\ \alpha \vee (\beta \wedge \gamma) & \Rightarrow & (\alpha \vee \beta) \wedge (\alpha \vee \gamma) \end{array}$$

By C we denote the set consisting of all of the above rules. A lexicographic path ordering, in which $\neg \succ \vee \succ \wedge \succ \top \succ \bot$ and other connectives have higher precedence than $\neg$, can be used to prove termination of C. We emphasize that we are interested in the existence of normal forms (or "weak normalization," which is ensured by termination), but not in their uniqueness. Indeed, it is well-known that conjunctive normal forms are not unique. Also, formulas in conjunctive normal form may contain certain "redundancies." For example,

$$((A \vee A) \wedge (B \vee A)) \wedge ((A \vee \neg B) \wedge (B \vee \neg B))$$

is a conjunctive normal form of $(A \wedge B) \vee (A \wedge \neg B)$, which could be further simplified to

$$(A \wedge (B \vee A)) \wedge (A \vee \neg B).$$

These additional simplifications will be part of the transformation to standard clauses discussed below.

Disjunctive normal form is obtained by distributing conjunctions over disjunctions:

$$\begin{array}{rcl}(\alpha \vee \beta) \wedge \gamma & \Rightarrow & (\alpha \wedge \gamma) \vee (\beta \wedge \gamma) \\ \alpha \wedge (\beta \vee \gamma) & \Rightarrow & (\alpha \wedge \beta) \vee (\alpha \wedge \gamma)\end{array}$$

To prove termination, we only need to slightly modify the above lexicographic path ordering, so that $\wedge \succ \vee$, instead of $\vee \succ \wedge$.

6.5 Tautology Deletion and Subsumption

Let us next list a few more examples of techniques for eliminating redundancies that are common in resolution provers.

Tautology deletion:

$N, \Sigma \rhd N$ if Σ is a tautology

Subsumption:

$N, \Sigma, (\Sigma, \Delta) \rhd N, \Sigma$

This ground version of subsumption is a simple example of deletion: a clause Σ, Δ is obviously redundant in the presence of another clause Σ. The corresponding version of subsumption for clauses with variables is an essential part of most resolution provers.

Subsumption resolution:

$$N, (\Delta, L), (\Sigma, \overline{L}) \rhd N, (\Delta, L), \Sigma$$

if L and $\overline{L}$ are complementary literals and $\Delta \subseteq \Sigma$

This rule represents a deduction step (resolution) followed by (zero or more) simplification steps and a deletion step (subsumption):

$$\begin{array}{ll}N, (\Delta, L), (\Sigma, \overline{L}) & \\ \quad \rhd & N, (\Delta, L), (\Sigma, \overline{L}), (\Delta, \Sigma) \\ \quad \rhd^* & N, (\Delta, L), (\Sigma, \overline{L}), \Sigma \\ \quad \rhd & N, (\Delta, L), \Sigma\end{array}$$

It (or rather its analogue for clauses with variables) is quite useful in practice. Let us emphasize that the implicit deduction step is by resolution, but not necessarily by *ordered* resolution, as the resolved atom need not be maximal in the premises. This is one of many examples where inferences are applied that are not necessary for reasons of completeness, but are nonetheless very useful in that they enable certain simplifications or deletions.

6.6 A Spectrum of Strategies

The theory developed so far provides us with a large spectrum of resolution-based theorem proving methods. Its particular instances are identified by various parameters, including the specific ordering, selection function, partial substitution strategy, simplification inferences, and redundancy detection methods that one chooses to realize. The significance of this spectrum and the general completeness result that comes with it is more of theoretical than of direct practical value. It will serve us to investigate the theoretical properties of more specialized resolution methods that have proven to be useful in practice. For good reasons those are usually based on a restricted syntax such as standard clauses, super clauses, sums of products, constrained clauses, and are endowed with specific simplification techniques that can be efficiently implemented. We will investigate some of them in more detail in Section 7 below.

Nevertheless, some of the power of general resolution should be more exploited in new designs of implementations. At present, transformation of a problem into, say, clausal normal form is considered as trivial and separate preprocessing. This view is not entirely justified. The loss of structure that comes with such preprocessing is enormous. Existential quantification and equivalences (including definition hierarchies) are no longer visible. While the problem of adequately dealing with existential quantification is beyond the scope of the present paper, our theory does allow for treating many cases of equivalences by simplification rather than search. Research into efficient methods of intertwining general resolution, simplification on the level of general clauses, normalform transformation and subsequent efficient computation with normalforms should be given more emphasis in the future.

7 Basic Resolution Strategies

This section is devoted to the discussion of resolution techniques that depend on a particular normal form of general clauses. Resolution, when applied to clauses in normal form, can be more efficiently implemented. Conceptually, resolution on normal forms is (restricted) general resolution followed by normalization so as to avoid the generation of non-normal clauses. The two steps are usually integrated into one and presented in the form of specifically modified resolution inferences. The standard redundancy criterion $\mathcal{R}^{\succ}$ for general resolution is at the same time a redundancy criterion for resolution with normalization provided the normalform transformation preserves logical equivalence and is compatible with the clause ordering $\succ$. In fact, if the normalization Σ' of the conclusion Σ is entailed by certain other clauses Δ_i, then Σ will be entailed by the Δ_i and Σ'. Since inferences and normalization are monotone, Σ' will be smaller that the [negative] premise of the inference and hence satisfies the ordering restrictions of $\mathcal{R}^{\succ}$.

7.1 Standard Resolution

We have seen that the replacement of a formula by its conjunctive normal form,

$$N, (\Sigma, F) \rhd N, (\Sigma, F') \qquad \text{if } F \Rightarrow^!_{\mathsf{C}} F'$$

is a simplification step (for a suitable ordering). We may take this normalization of formulas a step further and eliminate conjunctions from general clauses as follows:

$$\begin{aligned} & N, (\Sigma, F \wedge G) \\ & \qquad \rhd N, (\Sigma, F \wedge G), (\Sigma, F) \\ & \qquad \rhd N, (\Sigma, F \wedge G), (\Sigma, F), (\Sigma, G) \\ & \qquad \rhd N, (\Sigma, F), (\Sigma, G) \end{aligned}$$

(a sequence of two deduction steps followed by a deletion). In a similar way we may eliminate disjunctions:

$$N, (\Sigma, F \vee G) \rhd N, (\Sigma, F, G)$$

In short, any finite negative matrix can be reduced to an equivalent *standard matrix*, in which all clauses are standard clauses. Finally, we may eliminate certain redundancies by applying the following simplification rules:

$$\begin{aligned} N, (\Sigma, L, L) &\quad \rhd \quad N, (\Sigma, L) \\ N, (\Sigma, A, \neg A) &\quad \rhd \quad N, \top \\ N, \top &\quad \rhd \quad N \end{aligned}$$

For standard clauses, each occurrence of an atom is either positive or negative. A "minimal" selection function selects at most one negative atom in any clause. We call selection functions of this form *strong selection functions*. If we apply simple resolution with strong selection to standard clauses and simplify the conclusion by P, we obtain the following inference rule:

Standard ordered resolution with strong selection:

$$\frac{C \vee A \vee \ldots \vee A \quad D \vee \neg A}{C \vee D}$$

> where (i) C contains only non-selected atoms not greater than or equal to A and (ii) the atom A is either selected by S in $D \vee \neg A$, or else $D \vee \neg A$ contains no selected atoms at all and A is maximal and non-positive in D.

(The actual non-clausal resolvent is $C \vee \bot \vee \ldots \vee \bot \vee D \vee \neg\top$, which can obviously be simplified to $C \vee D$. Substitution replaces exactly one arbitrarily chosen negative occurrence of A in the negative premise.) Self-resolution needs not be applied to standard clauses, as all atoms are either positive or negative in them (cf. Proposition 6.2).

The standard resolution rule is often decomposed into factoring and binary resolution inferences. By $\mathsf{R}^{\succ}_{S}$ we denote the inference system consisting of ordered factoring and (binary) ordered resolution with (strong) selection.

Positive ordered factoring:

$$\frac{C \vee A \vee A}{C \vee A}$$

where A is maximal in C and no atom in C is selected.

(Binary) ordered resolution with selection:

$$\frac{C \vee A \quad D \vee \neg A}{C \vee D}$$

where (i) $B \not\succeq A$, for any atom B in C,[3] and C contains no selected atoms, and (ii) the atom A is either selected by S in $D \vee \neg A$, or else $D \vee \neg A$ contains no selected atoms at all and A is maximal and non-positive in D.

The positive factoring inference is of course just the deductive part implicit in a simplification $N,\ C \vee A \vee A \rhd N,\ C \vee A$.

We obtain as corollary to Theorem 5.4:

Theorem 7.1 *Let S be any strong selection function and $\succ$ be a completable ordering. If a standard negative matrix N is saturated up to redundancy under $\mathsf{R}^{\succ}_{S}$, then N is unsatisfiable if and only if it contains a contradiction.*

We emphasize that the notion of saturation up to redundancy is flexible, and general, enough so as to cover "mixed" derivations, in which both non-clausal and standard resolution inferences appear.

For certain applications below negative factoring will be required:

Negative ordered factoring:

$$\frac{C \vee \neg A \vee \neg A}{C \vee \neg A}$$

where A is selected or C contains no selected atom and A is maximal in C.

[3] If $\succ$ is total the constraint is equivalent to $A \succ B$, for any B in C.

7.2 Positive Resolution

A (general) clause is called *positive*, if it is false in the *empty Herbrand interpretation* $I_\perp$ (in which all atoms are false). If a clause Σ is positive, no atom can be selected in it, that is, $S(\Sigma)$ must be empty for any selection function S. Conversely, if a clause is not positive, simply selecting all its non-positive atoms yields an admissible selection. In fact, if C is non-positive and is false in an interpretation I then I contains an atom A of C. Note that a positive clause cannot contain a negative occurrence of an atom. The converse is not true in general.

Positive resolution for general clauses is obtained through a selection function S with $S(\Sigma)$ empty, whenever Σ is positive, and $S(\Sigma)$ the set of all non-positive atoms in Σ, if Σ is non-positive. In that case, General Ordered Resolution with Selection specializes to this inference:[4]

General positive ordered resolution:

$$\frac{\Sigma[A] \quad \Delta[A]}{\Sigma[A/\bot], \Delta[A|\top]}$$

where (i) the atom A is maximal in Σ, (ii) the clause Σ is positive, and (iii) the atom A is non-positive in Δ.

Let $\mathsf{GP}^{\succ}$ denote the resolution calculus consisting of positive ordered resolution and positive ordered self-resolution (that is, ordered self-resolution from positive premises). As an immediate consequence of Theorem 5.4 we obtain:

Theorem 7.2 *Let $\succ$ be a completable ordering. If N is saturated up to redundancy under $\mathsf{GP}^{\succ}$, then it is unsatisfiable if and only if it contains a contradiction.*

Unlike in the case of standard clauses where the negative premise in positive resolution is always non-positive (cf. below), for general positive resolution one cannot safely add the restriction that negative premises be non-positive. The inconsistent matrix consisting of the three atoms A, B, and C, and the positive clause $(\neg A \wedge C, \neg B \wedge C)$, with A the maximal atom, would otherwise be saturated.

7.3 Hyper-Resolution

The variants of resolution we have presented depend on two parameters, an ordering on formulas and a selection function. Let us discuss some specific settings for these parameters in the context of standard resolution.

[4]The polarity constraints that are represented by the Propositions 6.1 and 6.2 are attached as explicit restrictions to the inference.

Contrary to the general case, a standard clause is positive if it contains no negative atoms. We call a standard clause *negative* if it contains no positive literals. Let S be a selection function which selects, in any non-positive clause, exactly one negative atom. Ordered resolution with this type of selection is the following inference rule:

Positive ordered resolution:

$$\frac{C \vee A \quad D \vee \neg A}{C \vee D}$$

where (i) C is a positive clause and A is maximal in C, and (ii) the atom A is selected by S in $D \vee \neg A$.

This inference rule is of course also a special case of general positive ordered resolution.

If we choose a selection function that selects the sequence of all negative atoms in a clause, and if we choose to replace any of their occurrences, we get the following inference:

Ordered resolution with maximal selection:

$$\frac{C_1 \vee A_1 \quad \ldots \quad C_n \vee A_n \quad D_N \vee D_P}{C_1 \vee \ldots \vee C_n \vee D_P}$$

where (i) the clauses $C_1, \ldots, C_n$ and D_P are all positive, (ii) the C_i do not contain any of the atoms A_j, (iii) D_N is a negative clause containing the literals $\neg A_1, \ldots, \neg A_n$, and only those literals, and (iv) the A_i are maximal in the C_i, for all i.

The restriction (ii) has been added to make the polarity results of Proposition 6.1 explicit, assuming that ordered positive factoring takes care of clauses with multiple positive occurrences of maximal atoms.

Resolution with maximal selection is strongly related to hyper-resolution (Robinson 1965*a*): Let $C_1 \vee A_1, \ldots, C_n \vee A_n$ be positive clauses, where A_i is maximal in C_i, for all i. If there exist clauses $D_1, \ldots, D_{n+1}$, such that D_{i+1} is an ordered resolvent between C_i and D_i on A_i, and D_{n+1} is positive, then

$$\frac{C_1 \vee A_1 \quad \ldots \quad C_n \vee A_n \quad D_0}{D_{n+1}}$$

is called a *(positive) hyper-resolution* inference. The first n premises are called the *electrons*, the last premise the *nucleus* of the inference. Since all premises are standard clauses, the final (positive) clause D_{n+1} (but not the intermediate clauses $D_1, \ldots, D_n$) is independent of the order in which the electrons are listed.

Consider an inference by ordered resolution with maximal selelection and take D_i to be the simplified version of $(C_1, \ldots, C_i, D_N[A_1/\top, \ldots, A_i/\top], D_P)$, $0 \leq i \leq n$. Then D_{i+1} is a simplified version of the conclusion of the resolution inference on A_i with premises $C_i \vee A_i$ and D_i. While ordered resolution with maximal selection, therefore, is evidently a hyper-resolution inference, there are hyper-resolution inferences, such as

$$\frac{A_1 \vee A_2 \quad A_2 \vee A_3 \quad \neg A_1 \vee \neg A_2 \vee A_4}{A_2 \vee A_3 \vee A_4}$$

that are not ordered resolution inferences with maximal selection. (A_2 should not occur in the first premise.) In other words, ordered resolution inference with maximal selection is a more restrictive inference schema than hyper-resolution.

We should also point out that a resolution inference with maximal selection is redundant if any of the "intermediate resolvents" Γ_i is redundant (cf. Proposition 6.5). This answers an open question in (Wos 1988).

7.4 Consolution

A standard negative matrix $\bigwedge_{i \in I} C_i$ is valid if, and only if, each clause C_i with $i \in I$ contains a complementary pair of literals. If the matrix is also finite, its validity can therefore be checked easily. A finite positive matrix

$$\bigvee_{1 \leq i \leq m} \bigwedge_{1 \leq j \leq n_i} L_{i,j}$$

is inconsistent if, and only if, the negative matrix

$$\bigwedge_{1 \leq i \leq m} \bigvee_{1 \leq j \leq n_i} \overline{L}_{i,j}$$

is valid, where $\overline{L}_{i,j}$ is complementary to $L_{i,j}$.

One way of checking the validity of a positive matrix is by transforming it into an equivalent negative matrix. (Similarly, we may check whether a negative matrix is inconsistent by transforming it into an equivalent positive one.) This observation is the basis of a calculus called *consolution* Eder (1991). We describe (the propositional version of) this calculus by rewrite rules (in terms of formulas rather than matrices). Let AC be the set of the set of two-way rules

$$\begin{aligned}
(\alpha \wedge \beta) \wedge \gamma \quad &\Leftrightarrow \quad \alpha \wedge (\beta \wedge \gamma) \\
\alpha \wedge \beta \quad &\Leftrightarrow \quad \beta \wedge \alpha \\
(\alpha \vee \beta) \vee \gamma \quad &\Leftrightarrow \quad \alpha \vee (\beta \vee \gamma) \\
\alpha \vee \beta \quad &\Leftrightarrow \quad \beta \vee \alpha
\end{aligned}$$

and D be the rewrite system consisting of all generalized distributivity rules,

$$(\alpha_1 \wedge \ldots \wedge \alpha_m) \vee (\beta_1 \wedge \ldots \wedge \beta_n) \Rightarrow \bigwedge_{1 \leq i \leq m, 1 \leq j \leq n} \alpha_i \vee \beta_j$$

plus the rules

$$\alpha \vee \neg\alpha \Rightarrow \top \qquad \alpha \vee \alpha \Rightarrow \alpha$$

plus the rewrite rules from P for eliminating the constants $\top$ and $\bot$. Consolution corresponds to normalization by D/AC.

For example, we have

$$\begin{array}{lll} (A \wedge B) \vee (\neg A \wedge B) \vee \neg B & & \\ & \rhd & [(A \vee \neg A) \wedge (B \vee \neg A) \wedge (A \vee B) \wedge (B \vee B)] \vee \neg B \\ & \rhd^+ & [(B \vee \neg A) \wedge (A \vee B) \wedge B] \vee \neg B \\ & \rhd & (B \vee \neg A \vee \neg B) \wedge (A \vee B \vee \neg B) \wedge (B \vee \neg B) \\ & \rhd^+ & \top \end{array}$$

That is, the formula $(A \wedge B) \vee (\neg A \wedge B) \vee \neg B$ is valid.

In Eder's terminology, a dual clause is called a path. A path is called complementary if two of its literals are complementary. A multiset of paths, i.e., a positive matrix K, can be shortened to another positive matrix M, if there exists a surjective mapping s from K to M such that for any P in K, $s(P)$ is a subset of P. For instance, $C \vee D$ is a shortening of $(C \wedge D) \vee (C \wedge \neg A) \vee (A \wedge D)$. A positive matrix implies any of its shortenings. Consolution from two standard clauses $C \vee A$ and $D \vee \neg A$ is the inference consisting of (i) conjuncting the premises into $(C \vee A) \wedge (D \vee \neg A)$, (ii) applying distributivity, yielding $(C \wedge D) \vee (C \wedge \neg A) \vee (A \wedge D) \vee (A \wedge \neg A)$, (iii) eliminating complementary paths, giving, $(C \wedge D) \vee (C \wedge \neg A) \vee (A \wedge D)$, and, finally, (iv) shortening the result arbitrarily. As $C \vee D$ is one of the possible shortenings, resolution can be considered an instance of consolution. Despite this fact, consolution is more closely related to the connection or matings method (Andrews 1981, Bibel 1981) and to semantic tableau (cf. Section 10.2) than to resolution. If taken literally, all possible shortenings of paths have to be enumerated by the consolution inference. That refutational completeness is achieved by just considering the specific shortening $C \vee D$ is an insight which the theory of resolution provides.

7.5 Boolean Ring-Based Methods

Let AC be the set of (two-way) rewrite rules

$$\begin{array}{rcl} (\alpha \wedge \beta) \wedge \gamma & \Leftrightarrow & \alpha \wedge (\beta \wedge \gamma) \\ \alpha \wedge \beta & \Leftrightarrow & \beta \wedge \alpha \\ (\alpha \oplus \beta) \oplus \gamma & \Leftrightarrow & \alpha \oplus (\beta \oplus \gamma) \\ \alpha \oplus \beta & \Leftrightarrow & \beta \oplus \alpha \end{array}$$

and BR the set of rewrite rules

$$\begin{array}{rcl}
\neg\alpha & \Rightarrow & \alpha \oplus \top \\
\alpha \vee \beta & \Rightarrow & (\alpha \wedge \beta) \oplus (\alpha \oplus \beta) \\
\alpha \wedge \bot & \Rightarrow & \bot \\
\alpha \wedge \top & \Rightarrow & \alpha \\
\alpha \wedge \alpha & \Rightarrow & \alpha \\
\alpha \oplus \bot & \Rightarrow & \alpha \\
\alpha \oplus \alpha & \Rightarrow & \bot \\
(\alpha \oplus \beta) \wedge \gamma & \Rightarrow & (\alpha \wedge \gamma) \oplus (\beta \wedge \gamma)
\end{array}$$

All of these rules describe logical equivalences. Furthermore, the rewrite system BR/AC terminates and the corresponding normal forms, called *BR-normal forms*, are unique up to equivalence under AC (Hsiang 1985). We denote by $\Phi(F)$ a BR-normal form of F.

Normal forms may also be represented as (possibly empty) *sums*

$$P_1 \oplus P_2 \oplus \cdots \oplus P_n$$

of (pairwise different, possibly empty) *products* of (pairwise different) atoms

$$P_i = A_{i,1} \ldots A_{i,n_i}$$

where a product $A_1 A_2 \ldots A_n$ represents the formula

$$(A_1 \wedge (A_2 \wedge \cdots (A_{n-1} \wedge A_n) \cdots)$$

and a sum $P_1 \oplus P_2 \oplus \cdots \oplus P_n$ the formula

$$(P_1 \oplus (P_2 \oplus \cdots (P_{n-1} \oplus P_n) \cdots),$$

where empty products denote $\top$ and empty sums denote $\bot$. We often need to single out a specific atom and write

$$AP_1 \oplus \ldots \oplus AP_m \oplus Q_1 \oplus \ldots \oplus Q_n$$

with the understanding that none of the products $P_1, \ldots, P_m, Q_1, \ldots, Q_n$ contains A. We often use $AP \oplus Q$ as a short form for such a sum of products.

We will now design a refutationally complete calculus for formulas in BR-normal form. Our method is straightforward: we take a general resolution inference, the premises of which are in BR-normal form, and try to express its conclusion by an equivalent BR-normal form. Our notion of redundancy is powerful enough, so that the derived inference system is guaranteed to be refutationally complete.

Consider a general ordered resolution inference

$$\frac{AP \oplus Q \quad AP' \oplus Q'}{(\bot \wedge P) \oplus Q, (\top \wedge P') \oplus Q'}$$

in which both premises are sums of products in normal form. The disjunction

$$((\bot \wedge P) \oplus Q) \vee ((\top \wedge P') \oplus Q'),$$

which is logically equivalent to the resolvent, can be simplified to

$$Q \vee (P' \oplus Q').$$

Eliminating the disjunction symbol from the latter formula we get

$$Q(P' \oplus Q') \oplus Q \oplus (P' \oplus Q')$$

which is equivalent to

$$(Q \oplus \top)(P' \oplus Q') \oplus Q.$$

These considerations lead to the following inference rules:

Ordered BR-resolution:

$$\frac{AP \oplus Q \qquad AP' \oplus Q'}{\Phi((Q \oplus \top)(P' \oplus Q') \oplus Q)}$$

where both premises are BR-normal forms with maximal atom A.

BR-self-resolution:

$$\frac{AP \oplus Q}{\Phi(PQ \oplus P \oplus Q)}$$

where the premise is a BR-normal form with maximal atom A.

By BR we denote the inference system of these rules. As these rules are specialized versions, for premises in BR-normal form, of the general resolution inferences (with subsequent normalization of the conclusions), the notion of redundancy is exactly the same as given in the Definition 4.8.

Theorem 7.3 *Let N be a set of formulas in BR-normal form. If N is saturated with respect to* BR *and $\mathcal{R}^{\succ}$, then N is inconsistent if and only if it contains a contradiction.*

Proof. If N is saturated under $\mathsf{BR}^{\succ}$, then it is also saturated under $\mathsf{O}^{\succ}$. More specifically, a general ordered resolution inference

$$\frac{AP \oplus Q \quad AP' \oplus Q'}{(\bot \wedge P) \oplus Q, (\top \wedge P') \oplus Q'}$$

is redundant whenever the corresponding ordered BR-resolution inference

$$\frac{AP \oplus Q \quad AP' \oplus Q'}{\Phi((Q \oplus \top)(P' \oplus Q') \oplus Q)}$$

is redundant, and an ordered self-resolution inference

$$\frac{AP \oplus Q}{(\bot \wedge P) \oplus Q, (\top \wedge P) \oplus Q}$$

is redundant whenever the corresponding BR-self-resolution inference

$$\frac{AP \oplus Q \oplus \top}{\Phi(PQ \oplus P \oplus Q)}$$

is redundant. □

An alternative to ordered BR-resolution is a form of critical pair computation for equations between sums of products of atoms. A sum $AP \oplus Q$ with maximal atom A is viewed as an equation $AP \oplus Q \approx \top$ and oriented into a rewrite rule $AP \Rightarrow Q \oplus \top$. Given another rewrite rule $AP' \Rightarrow Q' \oplus \top$ with the same maximal atom, a critical pair $P(Q' \oplus \top) \approx P'(Q \oplus \top)$ exists that can itself be represented by the polynomial corresponding to $P(Q' \oplus \top) \oplus P'(Q \oplus \top) \oplus \top$. Hence (simple) BR-superposition is the following inference:

Simple BR-superposition:

$$\frac{AP \oplus Q \qquad AP' \oplus Q'}{\Phi(P(Q' \oplus \top) \oplus P'(Q \oplus \top) \oplus \top)}$$

where both premises are BR-normal forms with maximal atom A.

This inference rule is sound. In addition, we have

$$A,\, AP \oplus Q \models P \leftrightarrow (Q \oplus \top)$$

therefore also

$$\begin{aligned} A,\, AP \oplus Q \ \models & \\ & P(Q' \oplus \top) \oplus P'(Q \oplus \top) \oplus \top \\ & \leftrightarrow \ (Q \oplus \top)(Q' \oplus \top) \oplus P'(Q \oplus \top) \oplus \top \\ & \leftrightarrow \ (Q \oplus \top)(Q' \oplus \top \oplus P') \oplus \top \\ & \leftrightarrow \ (Q \oplus \top)(P' \oplus Q') \oplus Q. \end{aligned}$$

The calculation shows that if A and $AP \oplus Q$ are true the equivalence of the conclusions of a BR-superposition inference and a BR-resolution inference with the same premises follows. For proofs of redundancy of resolution inferences, the resolved atom and the positive premises may be assumed (cf. Proposition 4.9) Hence, the redundancy of a BR-superposition inference implies the redundancy of the corresponding ordered BR-resolution inference (with the same premises). Denoting by BRS the inference system of BR-superposition and BR-self-resolution, we therefore obtain the following completeness result:

Theorem 7.4 *Let N be a set of formulas in BR-normal form. If N is saturated with respect to* BRS *and $\mathcal{R}^{\succ}$, then N is inconsistent if and only if it contains a contradiction.*

Let us next briefly describe how a positive variant of BR-resolution can be derived from general positive resolution. First note that a product $A_1 \ldots A_n$ is false in the interpretation $I_\perp$ (in which all atoms are false), unless it is the trivial product $\top$. Consequently, a formula in BR-normalform is false in $I_\perp$ if, and only if, it does not contain the trivial product $\top$.

Positive BR-resolution:

$$\frac{AP \oplus Q \qquad AP' \oplus Q'}{\Phi((Q \oplus \top)(P' \oplus Q') \oplus Q)}$$

where both premises are BR-normal forms, the first premise contains no trivial product $\top$, and A is the maximal atom in the first premise.

On the other hand, a product $A_1 \ldots A_n$ is always true in the interpretation $I_\top$ (in which all atoms are true). Thus, a formula in BR-normalform is false in $I_\top$ if, and only if, it contains an even number of products and is different from $\perp$. The negative variant of ordered BR-resolution is therefore of the following form:

Negative BR-resolution:

$$\frac{AP \oplus Q \qquad AP' \oplus Q'}{\Phi((Q \oplus \top)(P' \oplus Q') \oplus Q)}$$

where both premises are BR-normal forms, the first premise is a sum of an even number of products, and A is the maximal atom in the first premise.

The refutational completeness of positive and, hence, negative BR-resolution (together with positive and, respectively, negative BR-self-resolution) is an immediate consequence of Theorem 7.2. It is clear that variants of these inference rules similar to BR-superposition are also refutationally complete. Incidentally, the completeness of the latter inference systems was posed as an open problem by Zhang (1994), who introduced negative BR-resolution and also mentions its positive dual. In (Zhang 1994) the negative variant of BR-resolution is called the "odd strategy," which may sound odd, given that the inference is characterized by the syntactic restriction that the first premise consist of an *even* number of products. However, Zhang represents the formula $AP \oplus Q$ by an equation $\Phi(AP \oplus Q \oplus \top) \approx \bot$, so that a formula with an even number of products is turned into an equation, the left-hand side of which consists of an odd number of products.

A Boolean ring-based method for first-order theorem proving was first described by Hsiang (1985). This so-called "N-strategy" is closely related to (standard) negative resolution.[5] The method applies to equations $A \approx \bot$ where A is a sum of products obtained from the negation of a given *clause*. That is, the initial formulas are assumed to be clauses, the negations of which are translated to sums of products. For instance, the negation of a negative clause $\neg P \vee \neg Q$ is represented by an equation $PQ \approx \bot$ (called an "N-rule") with a single product of atoms on the left-hand side. The N-strategy is a resolution method with the restriction that one of the premises of each inference be an N-rule. Thus, standard negative resolution is a special case. The N-strategy also allows for simplification by rewriting whereby equations may be transformed so that they no longer represent single standard clauses. However, the method is only complete if rather severe restrictions are imposed on simplification (Zhang 1994). Thus, the N-strategy in essence more closely resembles standard negative resolution, whereas negative-BR-resolution is a true non-clausal method, as shown above.

There are also slightly different approaches that do not derive from non-clausal resolution, but where critical pair computations and other techniques from associative-commutative completion are directly applied to the rewrite system BR/AC and polynomial equations; see (Kapur & Narendran 1985) and (Bachmair & Dershowitz 1987) for details.

8 Refined Techniques for Defining Orderings and Selection Functions

The basic resolution strategies of the preceeding section can be refined employing more elaborate techniques of defining orderings and selection functions. A key technique in this regard are renamings and conservative extensions of the theory. Renamings will allow to also select positive literals. Conservative extensions of the theory provide more freedom in defining orderings. The definitions of the new symbols of such an extension will often have to be (at least conceptually) pre-saturated as a first step of a theorem proving process. This fact will lead us to generally consider refinements of resolution in the presence of a saturated theory.

[5] Various improvements of the original N-strategy have been proposed, e.g., (Müller & Socher-Ambrosius 1988)

8.1 Renaming and semantic resolution

There are several refinements of resolution that are essentially syntactic variants of inference systems described in the previous sections. In particular, that is the case for *semantic resolution* (Slagle 1967) and, hence, for *set-of-support resolution*. Since these inference systems were originally introduced for standard clauses, we will restrict our discussion in this section to this case.

Let I be a Herbrand interpretation.

Semantic resolution:

$$\frac{C \vee L \quad D \vee \overline{L}}{C \vee D}$$

where (i) L and $\overline{L}$ are complementary literals, (ii) $C \vee L$ is false in I, and (iii) L is the maximal literal in the first premise.

More specifically, we speak of a semantic resolution *with respect to (the interpretation) I*. Let again $I_\perp$ be the interpretation in which all atoms are false, and $I_\top$ the interpretation in which all atoms are true. Semantic resolution with respect to $I_\perp$ corresponds to positive resolution;[6] semantic resolution with respect to $I_\top$ has been called *negative resolution*. Another well-known refinement of resolution that is covered by the restrictions of semantic resolution is *set-of-support resolution* (Wos, Robinson & Carson 1965): if N is a satisfiable matrix (the *theory*), we say that a resolution inference

$$\frac{C \vee L \quad D \vee \overline{L}}{C \vee D}$$

obeys the set-of-support restriction for N if at most one premise is a clause in N.

Positive and negative resolution are dual to each other in that positive resolution is based on the minimal (in a set-theoretic sense) Herbrand interpretation $I_\perp$, whereas negative resolution is based on the maximal Herbrand interpretation $I_\top$. It turns out that semantic resolution with respect to an interpretation I may be seen as a syntactic variant of semantic resolution with respect to any other interpretation I'. That is, the key aspects of semantic resolution can be captured syntactically, via renaming of literals. We outline how semantic resolution may be mapped to positive resolution, so that our completeness results are applicable.

Let $\mathcal{L}$ be a propositional language with atoms are $A_1, A_2, \ldots$ and $\mathcal{L}'$ be a propositional language (disjoint to $\mathcal{L}$) with atoms $A'_1, A'_2, \ldots$. Injective mappings ρ from $\mathcal{L}$ to the literals over $\mathcal{L}'$ are called *renamings* if, in addition, for no A'_i in $\mathcal{L}'$ both A'_i and $\neg A'_i$ are in the range of ρ. Renamings are

[6]For simplicity, we disregard selection in the negative premise.

homomorphically extended to clauses and matrices. assuming that expressions of the form $\neg\neg A$ are simplified to A. If I' is a Herbrand interpretation over $\mathcal{L}'$ and ρ a renaming of $\mathcal{L}$ into $\mathcal{L}'$ then by $\rho(I')$ we denote the Herbrand interpretation over $\mathcal{L}$ such that A is in $\rho(I')$ if and only if $\rho(A)$ is true in I'. I' is a model of a matrix $\rho(N)$ if and only if $\rho(I')$ is a model of N.

If I is a Herbrand interpretation for $\mathcal{L}$, then the *renaming ρ_I induced by I* is the renaming defined as follows: if $A_i \in I$, then $\rho_I(A_i) = \neg A'_i$ and if $A_i \notin I$, then $\rho_I(A_i) = A'_i$. In other words, ρ_I maps atoms in $\mathcal{L}$ that are false in I to positive literals in $\mathcal{L}'$, and atoms that are true in I to negative literals. Therefore, $\rho_I(I'_\perp) = \rho_I(\emptyset) = I$. If $\succ$ is an admissible ordering on expressions in $\mathcal{L}$, we define a corresponding ordering $\succ'$ for $\mathcal{L}'$ by: $E \succ E'$ if, and only if, $\rho_I(E) \succ' \rho_I(E')$. $\rho_I(C)$ is true in $I'_\perp$', that is, is a clause with a negative literal, if and only if C is true in I. With these definitions, whenever

$$\frac{C \vee L \quad D \vee \overline{L}}{C \vee D}$$

is a semantic resolution inference with respect to I, then

$$\frac{\rho_I(C \vee L) \quad \rho_I(D \vee \overline{L})}{\rho_I(C \vee D)}$$

is a positive resolution inference. In short, semantic resolution with respect to any interpretation I may be viewed as a syntactic variant of positive resolution.

Proposition 8.1 *Let N be a negative and consistent matrix of standard clauses. Then there exists a renaming ρ and a selection function S such that $\rho(N)$ is saturated up to redundancy by ordered resolution with respect to S and any ordering.*

Proof. If I is a model of N, choose ρ to be ρ_I and observe that no clause in $\rho(N)$ is positive. Then choose S in a way such that some atom is selected in each clause in $\rho(N)$. Since every clause in $\rho(N)$ must have a selected atom, no inference from premises in $\rho(N)$ exist. $\Box$

The significance of this proposition lies in the fact that it asserts the existence of a saturated presentation for any consistent theory, in particular establishing the refutational completenes of the set-of-support strategy.

8.2 Resolution with free selection

Recall that we have obtained positive resolution by choosing the selection function in a suitable way. For standard clauses it suffices to select, in any non-positive clause, some negative atom. The observations in Section 8.1 show that we may elude the limitation of selection to negative occurrences of atoms, by renaming literals so that every positive occurrence of an atom A we wish to select is turned into a negative occurrence of the renamed atom A'.

These results, however, are not quite sufficient to explain the refutational completeness of free selection for Horn clauses. A *free selection function* selects exactly one (or, more generally, at least one) arbitrary, positive or negative, atom in any (nonempty) clause. Then, *binary resolution with free selection* is restricted to inferences which resolve an atom only if it is selected in *both* premises. Completeness results for binary resolution with free selection have been obtained by Lynch (1997) and de Nivelle (1996). We present a simple proof of this fact that is based on proof transformations on resolution proof trees.

Let N be a matrix of Horn clauses. (Remember that Horn clauses with empty heads are admitted and are written in the form $\neg A_1 \vee \ldots \vee \neg A_k \vee \perp$, with atoms A_i.) A proof of an atom A [or of $A = \perp$] from N is an ordered tree with these properties: (i) any node different from the root is labeled by an atom; (ii) its root is labeled A; and (iii) for any node v labeled B in the tree there exists a clause $C = \neg A_1 \vee \ldots \vee \neg A_k \vee B$ in N such that v has exactly k descendants labeled $A_1, \ldots, A_k$. (In that case we say that the clause C has been applied at v.)

Theorem 8.2 *Let N be a matrix of Horn clauses that contains all resolvents by binary resolution inferences with free selection from N. If $N \models \perp$ then N contains $\perp$.*

Proof. If $N \models \perp$ then there exists a proof of $\perp$ from N. Let be a minimal (with respect to the number of nodes) such proof be given. If the proof simply consists of $\perp$ we are finished. Otherwise we propose a construction that will leave us with a contradiction, showing that the minimal proof cannot be nontrivial. Suppose the proof has more than one node. Then we may consider any path $v_1, \ldots, v_k$ from the root v_1 downward in the tree such that, for any $1 \leq i < k$, if $C = \neg A_1 \vee \ldots \vee \neg A_k \vee B$ is the clause that is applied at node v_i and if v_{i+1} is the j-th descendant of v_i, then A_j is selected in C. If the path is chosen such that it has maximal length, that is, cannot be extended any further, then $k \geq 2$. In fact, since the clause applied at the root of the proof has an empty head $\perp$, it must have a negative atom selected, and we may choose the corresponding descendant of v_1 for v_2. Moreover in the clause D which is applied at v_k the positive, but no negative, atom is selected, as otherwise the path would be extensible. We infer that a resolution inference is possible between D and the clause D' applied at v_{k-1} which obeys the restrictions about selection. (In fact, the atom at v_k is selected both in D and D'.) As the resolvent is contained in N we might use it to construct a proof of $\perp$ which has one node less than the proof that we started with. This is a contradiction. □

A consequence of this theorem is the completeness of SLD-resolution (Kowalski 1974) for Prolog: as a particular free selection function one chooses one that selects, respectively, for program clauses their heads, and, for negative goal clauses, any of their negative literals.

One can easily see that resolution with free selection (together with unrestricted positive and negative factoring) is generally incomplete for non-Horn clauses. For an example, consider the inconsistent matrix

$$A \vee \underline{B},\ \underline{A} \vee \neg B,\ \neg \underline{A} \vee B,\ \neg A \vee \neg \underline{B}$$

with selection as indicated by underlining. The two inferences possible derive $B \vee \neg \underline{B}$ and $A \vee \neg \underline{A}$, respectively. Even if these tautologies are not eliminated, selecting as indicated only allows to derive clauses which are present already. de Nivelle (1996) has shown, however, that free selection for full clauses may cause incompleteness only when every resolution-based proof of inconsistency requires at least one step of factoring. That is the case in particular for every inconsistent set of binary clauses. Resolution between two binary clauses produces again a binary clause, and only by factoring steps clauses may become shorter.

SL-resolution (Kowalski & Kuehner 1971) is a resolution strategy for full clauses which also employs a rather liberal selection strategy that cannot be directly justified within the essentially semantic framework described until now. SL-resolution is a refinement of set-of-support resolution where arbitray, positive or negative atoms may be selected, provided they have been introduced by a theory clause premise of a previous resolution step. SL-resolution is closely related to model elimination (Loveland 1969) and hence to semantic tableau. In Sections 10.2 and 10.3 we shall briefly describe how to generalize the linear theorem proving derivations of the Section 4.2 to derivation trees. This will allow us to model semantic tableau and refinements such as model elimination in our framework. We shall in particular see that those aspects of selection that cannot be modeled on the semantic level of partial interpretations can be justified on the meta-level of derivations.

To keep this paper within reasonable bounds, this proof-theoretic dimension of resolution cannot be explored in much detail in this paper. The proof of Theorem 8.2 indicates that one can define redundancy criteria for resolution which are based on proof orderings. These could be much more refined than the simple node count that we have used there. Our standard redundancy criterion corresponds to a proof ordering where proofs are compared by simply comparing the sets of assumptions they apply. This gives a quasi-ordering where all proofs that use the same assumptions would become equivalent. More fine-grained orderings may provide a justification of more sophisticated redundancy elimination techniques than we can deal with in the present paper. On the other hand it appears that proof-theoretic completeness proofs for general ordered resolution with selection are technically much more complex and, moreover, cannot easily be extended to refinements of resolution such as chaining or superposition. In (Bachmair 1989) completeness proofs for certain calculi of ordered resolution and paramodulation have been given by means of proof transformations. Proof transformations for a superposition-type calculus for Horn clauses with equality have been explored in (Ganzinger 1991).

8.3 Conservative Extensions

A straightforward transformation to clausal form or conjunctive normal form—for instance, via normalization with the rewrite system C—may exponentially increase the size of a formula. Fortunately, there are other transformation schemes that preserve the consistency of a formula, but avoid an exponential increase in size. They are based on the concept of the extension of a language by new predicate symbols and corresponding definitions.

More formally, let $N[F]$ be a matrix containing a subformula F and L be a literal P or $\neg P$, where P is an atom not occurring in N. We say that the matrix N', M is a *regular extension* of $N[F]$ (by L) if N' is obtained from N by replacing one or more occurrences of F by L and if M is logically equivalent to $L \leftrightarrow F$. Similarly, we speak of a *positive* (resp., *negative*) *extension* if only positive (resp., negative) occurrences[7] of F in N are replaced and if M is logically equivalent to $L \rightarrow F$ (resp., $F \rightarrow L$). Finally, we say that K is a *(conservative) extension* of N if it is obtained from N by a sequence of finitely many (but at least one) regular, positive and/or negative extensions.

Proposition 8.3 *If K is an extension of N, then K is consistent if, and only if, N is consistent.*

Proof. It is sufficient to show the assertion for every regular, positive, and negative extension.

Let I be a model of $N[F]$ and L be a literal P or $\neg P$, such that P is not contained in N. If L and F have the same truth value in I, then I is a model of any (regular, positive, or negative) extension of $N[F]$ by L. If L and F have different truth values in I, define I' to be $I \cup \{P\}$, if $P \notin I$; and $I \setminus \{P\}$, if $P \in I$. Then L and F have the same truth value in I' (since P does not occur in F), and I' is a model of any extension of $N[F]$ by L. We have thus shown that the consistency of N implies the consistency of all of its regular, positive, and negative extensions.

If I is a model of a regular extension N', M of $N[F]$ by L, then L and F have the same truth value in I and, hence, N' and $N[F]$ also have the same

[7]We speak of a positive [negative] occurrence of F in matrix N if that occurrence of F is within a clause C of N in which F is a positive [negative] subformula.

truth value in I. Since I is a model of N', it has to be a model of $N[F]$ as well.

If I is a model of a positive extension N', M of $N[F]$ by L, then I is a model of N' and of $L \to F$. Again, if L and F have the same truth value in I, then $N[F]$ is true in I. If L is true in I, then F must also be true in I, for otherwise the implication $L \to F$ would be false. Thus, if L and F have different truth values in I, then L is false in I and F is true in I. Since N' results from replacing positive occurrences of F in N, for any such occurrence $C[F]$ within a clause C of N the clause $C[F/\top]$ is a tautology, which implies that $C[F]$ is true in I. Clauses in N in which F is not replaced also occur in N' and, hence, are true in I by assumption.

The case of negative extensions is handled in a similar way. In sum, the consistency of any regular, positive or negative extension of N implies that N is also consistent. $\Box$

We illustrate the use of the extension principle by showing how any formula (or any finite matrix, for that matter) can be converted to an equivalent standard matrix so that the size increases only by a constant factor, cf., Tseitin (1970).

Let $E_P(L \wedge L')$ denote the standard matrix (of three clauses)

$$\neg P \vee L, \neg P \vee L', P \vee \overline{L} \vee \overline{L'},$$

where L and $\overline{L}$, and also L' and $\overline{L'}$, are complementary literals. Similarly, let $E_P(L \vee L')$ be the standard matrix

$$\neg P \vee L \vee L', P \vee \overline{L}, P \vee \overline{L'};$$

$E_P(L \to L')$ the matrix

$$\neg P \vee \overline{L} \vee L', P \vee L, P \vee \overline{L'};$$

and $E_P(L \leftrightarrow L')$ the matrix

$$\neg P \vee \overline{L} \vee L', \neg P \vee L \vee \overline{L'}, P \vee L \vee L', P \vee \overline{L} \vee \overline{L'}.$$

These matrices satisfy the following logical equivalences:

$$\begin{array}{rcl}
E_P(L \wedge L') & \equiv & P \leftrightarrow (L \wedge L') \\
E_P(L \vee L') & \equiv & P \leftrightarrow (L \vee L') \\
E_P(L \to L') & \equiv & P \leftrightarrow (L \to L') \\
E_P(L \leftrightarrow L') & \equiv & P \leftrightarrow (L \leftrightarrow L')
\end{array}$$

Let now N be a finite matrix. For simplicity, we assume that all formulas in N are in negation normal form. (Transformation to negation normal form increases the size of a formula by a constant factor only.) If N is not a standard matrix, it must contain a subformula $L \circ L'$, where $\circ$ is one of the connectives $\wedge$, $\vee$, $\rightarrow$ or $\leftrightarrow$ and L and L' are literals. Then $N[P_{L \circ L'}], E_{P_{L \circ L'}}(L \circ L')$ is an extension of $N[L \circ L']$, where $P_{L \circ L'}$ is a new predicate symbol. Each extension step eliminates at least one occurrence of a binary connective, so that we eventually end up with a standard matrix that is consistent if, and only if, the initial matrix N is consistent. Since in the worst case each occurrence of a logical connective in the initial formula has to be replaced by a new atom and a matrix of at most four clauses, each with no more than three literals, the size of the initial matrix may increase only by a constant factor.

Plaisted & Greenbaum (1986) have presented a refinement of this transformation scheme in which the polarities of abbreviated formulas $L \circ L'$ are considered so that for a positive [negative] $L \circ L'$ a positive [negative] extension by $P_{L \circ L'}$ is generated. They also discuss how to automatically extend the ordering to the new $P_{L \circ L'}$-atoms in a way such that predicates that abbreviate small formulas are preferred for ordered inferences.

8.4 Lock resolution

Extension allows us to obtain some interesting variations of resolution inference systems. For instance, lock resolution (Boyer 1971) can essentially be encoded by positive hyper-resolution.

Lock resolution is applied to clauses in which each occurrence of a literal has been assigned an integer, called a *lock index*. For example, in the following matrix N_0 of four clauses,

$$\begin{array}{lcr} {}_1A & \vee & {}_2B, \\ {}_3\neg A & \vee & {}_4\neg B, \\ {}_5B & \vee & {}_6\neg A, \\ {}_7\neg B & \vee & {}_8A \end{array}$$

each literal occurrence has a unique index, but in general different literal occurrences may be assigned the same index. The lock restriction states that only literals with a minimal index must be resolved. More formally, we have the following inference rules:

Lock resolution:

$$\frac{C \vee {}_iA \quad {}_j\neg A \vee D}{C \vee D}$$

where no literal in C has a smaller index than i, and no literal in D has a smaller index than j.

Lock factoring:

$$\frac{C, {}_iA, {}_jA}{C, {}_iA}$$

where no literal in C has a smaller index than i.

For example,

$$\frac{{}_1A \vee {}_2B \qquad {}_3\neg A \vee {}_4\neg B}{{}_2B \vee {}_4\neg B}$$

is a lock resolution inference, but

$$\frac{{}_1A \vee {}_2B \qquad {}_5B \vee {}_6\neg A}{{}_2B \vee {}_5B}$$

is not.

Let $N = C_1, \ldots, C_n$ be a finite matrix of standard clauses (with lock indices). For each clause

$$C_i = {}_{l_{i,1}}A_{i,1} \vee \cdots \vee {}_{l_{i,k_i}}A_{i,k_i} \vee {}_{l_{i,k_i+1}}\neg B_{i,k_i+1} \vee \cdots \vee {}_{l_{i,k_i+m_i}}\neg B_{i,k_i+m_i}$$

let C_i' be the (renamed) clause

$$C_i = P_{i,1} \vee \cdots \vee P_{i,k_i} \vee P_{i,k_i+1} \vee \cdots \vee P_{i,k_i+m_i}$$

where the $P_{i,j}$ are predicate constants not occurring in N;[8] and let M_i be the matrix of all clauses $\neg P_{i,j} \vee A_{i,j}$, where $1 \leq j \leq k_i$, and $\neg P_{i,k_i+l} \vee \neg B_{i,k_i+l}$, where $1 \leq l \leq m_i$. We also say that $P_{i,j}$ *encodes* $A_{i,j}$ or $\neg B_{i,j}$, respectively. Finally, let N' be the (renamed) matrix $C_1', \ldots, C_n'$ and M be the matrix $M_1, \ldots, M_n$. The clauses in M are called *definitions*.

Note that a clause $\neg P_{i,j} \vee A_{i,j}$ is logically equivalent to the implication $P_{i,j} \rightarrow A_{i,j}$, and $\neg P_{i,j} \vee \neg B_{i,j}$ is logically equivalent to $P_{i,j} \rightarrow \neg B_{i,j}$. Thus, N', M is an extension of N.

For example, from the matrix N_0 above we get a renamed matrix

$$\begin{array}{lcl} P_{1,1} & \vee & P_{1,2} \\ P_{2,1} & \vee & P_{2,2} \\ P_{3,1} & \vee & P_{3,2} \\ P_{4,1} & \vee & P_{4,2} \end{array}$$

where the predicate constants $P_{i,j}$ are defined by this matrix M:

$$\begin{array}{lcl@{\qquad\qquad}lcl} \neg P_{1,1} & \vee & A & \neg P_{1,2} & \vee & B \\ \neg P_{2,1} & \vee & \neg A & \neg P_{2,2} & \vee & \neg B \\ \neg P_{3,1} & \vee & B & \neg P_{3,2} & \vee & \neg A \\ \neg P_{4,1} & \vee & \neg B & \neg P_{4,2} & \vee & A \end{array}$$

Let now $\succ$ be an ordering in which (i) $A_{i,j} \succ P_{i,j}$ and $B_{i,j} \succ P_{i,j}$, for all i, j, and (ii) $P_{i,j} \succ P_{i',j'}$ if, and only if, the lock index $l_{i,j}$ associated with the literal encoded by $P_{i,j}$ is smaller than the lock index $l_{i',j'}$ associated with

[8] We assume there is a sufficient supply of symbols $P_{i,j}$ in the given propositional language.

the literal encoded by $P_{i',j'}$. We then saturate the matrix M under ordered resolution. This results, with the given ordering, in the elimination of the "old" atoms $A_{i,j}$, that is, the result is a matrix M, K, where K consists of all two-element negative clauses $\neg P_{i,j} \vee \neg P_{k,l}$, such that $P_{i,j}$ and $P_{k,l}$ encode complementary literals. The clauses in K are called *connections*.

For the above example we obtain the following connections:

$$
\begin{array}{lcl@{\qquad\qquad}lcl}
\neg P_{1,1} & \vee & \neg P_{2,1} & \neg P_{1,2} & \vee & \neg P_{2,2} \\
\neg P_{1,1} & \vee & \neg P_{3,2} & \neg P_{1,2} & \vee & \neg P_{4,1} \\
\neg P_{3,1} & \vee & \neg P_{4,1} & \neg P_{4,2} & \vee & \neg P_{2,1} \\
\neg P_{3,1} & \vee & \neg P_{2,2} & \neg P_{4,2} & \vee & \neg P_{3,2}
\end{array}
$$

For instance, the connection $\neg P_{1,1} \vee \neg P_{3,2}$ indicates that in the original matrix N_0 the first literal in the first clause is complementary to the second literal in the third clause.

The matrix N', M, K is partially saturated in the sense that all inferences with premises from M (i.e., definitions) are redundant in this context. If we use a selection function that selects both literals in a connection, then any possible ordered resolution inference with this selection must be of the form

$$
\frac{C \vee P_{i,j} \quad D \vee P_{k,l} \quad \neg P_{i,j} \vee \neg P_{k,l}}{C \vee D}
$$

where $P_{i,j}$ is maximal in the first positive premise and $P_{k,l}$ is maximal in the second positive premise. The negative premise is a connection and the conclusion is again a positive clause. In other words, these are positive hyper-resolution inferences with connections as nucleus. The ordering restrictions guarantee that these hyper-resolution inferences encode lock resolution inferences. More precisely, if $C' \vee L$ denotes the clause obtained from $C \vee P_{i,j}$ by replacing each atom by the literal it encodes, and $D' \vee \overline{L}$ is obtained in the same way from $D \vee P_{k,l}$, then

$$
\frac{C' \vee L \quad D' \vee \overline{L}}{C' \vee D'}
$$

is a lock resolution inference. Conversely, each lock resolution inference is encoded by a positive hyper-resolution inference of the above form. In sum, there is a one-to-one correspondence between hyper-resolution inferences (with two renamed clauses as negative premises and a connection as positive premise) and lock resolution (on the original clauses).

For example, the hyper-resolution inference

$$
\frac{P_{3,1} \vee P_{3,2} \quad P_{4,1} \vee P_{4,2} \quad \neg P_{3,1} \vee \neg P_{4,1}}{P_{3,2} \vee P_{4,2}}
$$

encodes the lock resolution inference

$$\frac{{}_5B \vee {}_6\neg A \qquad {}_7\neg B \vee {}_8A}{{}_6\neg A \vee {}_8A}$$

It is well-known that lock resolution is not compatible with tautology deletion. For example, the two lock resolution inferences we have shown above are the only ones from premises in N_0. In each case, the conclusion is a tautology. If these inferences were regarded as redundant, then N_0 would be saturated up to redundancy, yet is inconsistent and contains no contradiction. This appears to contradict the fact that our completeness results cover redundancy. But redundancy, in our sense, has to be applied to encoded clauses and inferences and we can see from the example that the encoding, $P_{3,2} \vee P_{4,2}$, of the tautology ${}_6\neg A \vee {}_8A$ is not itself a tautology and therefore is not redundant.

8.5 The inverse method

The inverse method was proposed by Maslov (1964). Its basic inference rules are formulated in terms of a given set of (generalized) disjunctions of conjunctions of formulas. In our description of the method we follow Lifschitz (1989), where the method is formulated for disjunctions $G_1 \vee \ldots \vee G_k$ of conjunctions $G_i = L_{i1} \wedge \ldots \wedge L_{im_i}$ of literals L_{ij}. The disjunctions have been called "superclauses", and the conjunctions G_i, "superliterals" in (Lifschitz 1989). The negations $\neg G_i$, which are equivalent to standard clauses in the L_{ij}, will be denoted by $\overline{G}_i$. Given a set of input superclauses, an *S-clause* is any standard clause that is logically equivalent to a disjunction of negated superliterals $\overline{G}_i$ in the input. For example, given the input

$$\begin{array}{l} (P \wedge \neg Q) \vee (R \wedge T) \\ \neg P \vee Q \\ \neg R \end{array}$$

the input superliterals are the conjunctions

$$P \wedge \neg Q,\ R \wedge T,\ \neg P,\ Q,\ \neg R,$$

and their negations are the clauses

$$\neg P \vee Q,\ \neg R \vee \neg T,\ P,\ \neg Q,\ R\ .$$

Forming, for instance, the disjunction consisting of the negated first and third superliterals yields the clause

$$\neg P \vee Q \vee P$$

which is a tautology. They represent connections between complementary literals in the input superclauses.

The inverse method consists of the following two inference rules:

Type A inferences

$$\frac{}{C}$$

where C is any S-clause which is a tautology.

Type B inferences

$$\frac{E_1 \vee \overline{G}_1 \quad \ldots \quad E_k \vee \overline{G}_k}{E_1 \vee \ldots \vee E_k}$$

where $G_1 \vee \ldots \vee G_k$ is an input superclause, and where the premises are S-clauses.

Clearly, the conclusion of a type B inference is again an S-clause. In (Lifschitz 1989) factoring is built into the set notation for clauses.

We will show how to encode this standard version of the inverse method by positive hyper-resolution. The encoding will be similar to the encoding of lock resolution. Let N be a matrix $F_1, \ldots, F_n$, where each F_i is a disjunction $G_{i,1} \vee \ldots \vee G_{i,m_i}$ of conjunctions $G_{i,j}$ of literals. For each conjunction $G_{i,j}$ we introduce a propositional constant $P_{i,j}$ that does not occur in N, and denote by $M_{i,j}$ a standard matrix logically equivalent to $\neg P_{i,j} \rightarrow G_{i,j}$. By M we denote the matrix $M_{1,1}, \ldots, M_{n,m_n}$. Let C_i be the clause $\neg P_{i,1} \vee \ldots \vee \neg P_{i,m_i}$ and N' be the standard matrix $C_1, \ldots, C_n$. Then N', M is an extension of N. (Note that it is sufficient to introduce one constant $P_{i,j}$ for all occurrences of a formula $G_{i,j}$; we need not introduce different constants for different occurrences of $G_{i,j}$ in N.) In the example N' has the clauses

$$\begin{array}{lcl} \neg P_{1,1} & \vee & \neg P_{1,2} \qquad (1) \\ \neg P_{2,1} & \vee & \neg P_{2,2} \qquad (2) \\ \neg P_{3,1} & & \qquad\qquad\;\; (3) \end{array}$$

with the $\neg P_{i,j} \rightarrow G_{i,j}$ of the form

$$\begin{array}{lcl} \neg P_{1,1} & \rightarrow & P \wedge \neg Q \\ \neg P_{1,2} & \rightarrow & R \wedge T \\ \neg P_{2,1} & \rightarrow & \neg P \\ \neg P_{2,2} & \rightarrow & Q \\ \neg P_{3,1} & \rightarrow & \neg R \end{array}$$

and, hence, M the standard matrix

$$\begin{array}{lcr} P_{1,1} & \vee & P \\ P_{1,1} & \vee & \neg Q \\ P_{1,2} & \vee & R \\ P_{1,2} & \vee & T \\ P_{2,1} & \vee & \neg P \\ P_{2,2} & \vee & Q \\ P_{3,1} & \vee & \neg R \;\; . \end{array}$$

As can be observed from the example, in $\neg P_{i,j} \rightarrow G_{i,j}$ (the "definition" of the literal $\neg P_i, j)$, if $G_{i,j}$ is a conjunction $L^1_{i,j} \wedge \cdots \wedge L^{m_{i,j}}_{i,j}$, then the implication is logically equivalent to the matrix of binary clauses $P_{i,j} \vee L^1_{i,j}, \ldots, P_{i,j} \vee L^{m_{i,j}}_{i,j}$. Let $\succ$ be an admissible ordering in which all new atoms $P_{i,j}$ are smaller than all old atoms occurring in N. If we saturate M under ordered resolution, the result is a matrix M, K, where K is a matrix of positive clauses of the form $P_{i,j} \vee P_{i',j'}$. The clauses in K encode the tautologies that can be obtained by "Type A" inferences. In the example, K consists of the clauses

$$
\begin{array}{lcll}
P_{1,1} & \vee & P_{2,1} & \qquad\qquad (4)\\
P_{1,1} & \vee & P_{2,2} & \qquad\qquad (5)\\
P_{1,2} & \vee & P_{3,1} & \qquad\qquad (6).
\end{array}
$$

Let us also use a selection function that selects all negative literals in a clause. Then the non-redundant resolution inferences during saturation of N', M, K are positive hyper-resolution inferences of the form

$$\frac{D_1 \vee L_1 \ldots D_n \vee L_n \quad \neg L_1 \vee \cdots \vee \neg L_n}{D_1 \vee \cdots \vee D_n}$$

where the positive premises are positive clauses (initially from K) and the negative premise is from N'. The conclusion is again a positive clause. These inferences correspond to "Type B" inferences (The negative premise $\neg L_1 \vee \cdots \vee \neg L_n$ encodes one of the formulas in the original matrix N.) Conversely, any "Type B" inference can be translated into a positive hyper-resolution inference of this form. In short, we have established a one-to-one correspondence between the (standard version of the) inverse method and positive hyper-resolution. For refutational completeness, ordered factoring for positive clauses has to be added.

In the example one derives a contradiction by the following series of type B inferences:

(7)	$P_{1,2}$	[(6) into (3)]
(8)	$P_{2,1}$	[(4) and (7) into (1)]
(9)	$P_{1,1}$	[(5) and (8) into (2)]
(10)	$\perp$	[(7) and (9) into (1)]

Maslov's superclauses represent a particular (non-standard) clausal normal form. Specializing general resolution to superclauses would result in an inference

$$\frac{\Sigma, A \wedge G \quad \Delta, \neg A \wedge H}{\Sigma[A/\perp], \Delta[A|\top]}$$

which is related to (a sequence of two) type B inferences of the inverse method but different in the way multiple occurrences of the resolved atom A are handled.

8.6 Ordered Theory Resolution

Theory Resolution is concerned with specializing resolution to a specific (consistent) submatrix T of a given matrix, the *theory*. (The clauses not in T will be called *goal clauses*.) The concept was introduced by Stickel (1985). Instances of theory resolution include resolution modulo an equational theory E in which case syntactic unification is replaced by E-unification, or constraint resolution such as in (Bürckert 1990). A minimal requirement for any calculus of theory resolution is that no explicit inferences with the theory should be required. That is equivalent to requiring that T be saturated with respect to some particular instance of resolution. In Section 8.1 we have shown that, by renaming, one can always obtain a saturated presentation. Hence, set-of-support resolution may be viewed as an instance of theory resolution. The presentation of T which underlies set-of-support resolution is in a trivial way saturated: After renaming, none of the theory clauses is positive, and selecting their non-positive parts makes inferences syntactically impossible unless one of the premises is a goal clause. In particular, no non-trivial consequences of T are explicity represented.

If T is saturated by ordered resolution *without selelection* then more powerful theory resolution schemes can be obtained. Technically, the results in this section strictly extend both the results by Baumgartner (1992) and Bronsard & Reddy (1992). We will again restrict our presentation to the case of propositional standard clauses.

To facilitate notation we will at places employ signed atoms to denote literals: A sign is either "+" or "−", where $+A$ denotes the positive literal A while $-A$ denotes the negative literal $\neg A$. σ and τ denote signs. The complement $\overline{\sigma}$ of a sign σ is "−" if σ is "+", and "+", otherwise.

Let $\succ$ be a completable ordering. By $\succ_w$ we denote the partial ordering on clauses defined as $C \succ_w D$ if and only if for any atom B in D there exists an atom A in C such that $A \succ B$. We call $\succ_w$ the *weak extension* of $\succ$ to clauses. $\succ_w$ is well-founded, but in general not an admissible clause ordering. For $\succ_w$ only the maximal atoms matter. Clauses that have the same or uncomparable maximal atoms cannot be compared in $\succ_w$

We assume that T is saturated up to redundancy with respect to binary ordered resolution $\mathsf{R}^{\succ}$ (with positive *and* negative ordered factoring) *without selection* and redundancy criterion $\mathcal{R}^{\succ_w}$. That is we assume resolution ordering restrictions according to $\succ$ and ordering restrictions for redundancy according to $\succ_w$. In general $\mathcal{R}^{\succ_w}(N)$ is a proper subset of $\mathcal{R}^{\succ}(N)$, both for clauses and inferences, and for any matrix N.

In general, as $\succ$ may be partial, a theory clause can have more than one maximal atom. If $\succ'$ is a well-founded (not necessarily total) admissible extension of $\succ$, ordered theory resolution with respect to $\succ'$ is the following inference:

Ordered theory resolution

$$\frac{C_1 \vee \sigma_1 A_1 \quad \ldots \quad C_k \vee \sigma_k A_k}{C_1 \vee \ldots \vee C_k \vee D}$$

where the $C_i \vee \sigma_i A_i$ are $k \geq 1$ goal clauses and there exists a theory clause $\overline{\sigma}_1 A_1 \vee \ldots \vee \overline{\sigma}_k A_k \vee D$ such that (i) $B \not\succeq' A_i$ for any atom B in C_i, (ii) the A_i are pairwise incomparable under $\succ$, and (iii) for any atom B in D there exists an i such that $A_i \succ B$.

In essence, ordered theory resolution is hyper-resolution of goal clauses into a theory clause such that all the maximal atoms of the theory clause are resolved simultaneously. The non-maximal atoms in the theory clause remain as a (smaller) residuum. On the side of the goal clauses only maximal atoms participate in the inference, whereby the ordering can be any well-founded extension of the ordering $\succ$ with respect to which the theory is saturated.

Ordered theory resolution, together with ordered resolution and ordered factoring (both positive and negative), both restricted to goal clauses, is refutationally complete. We will briefly sketch how the completeness proof can be reduced to the completeness proof for general ordered resolution. That reduction will implicitly construct the redundancy criterion that can be associated with the theory resolution inference.

The key technique, again, is extension and renaming so that (i) the maximal atoms in theory clauses become negative literals which can be selected afterwards, and (ii) goal clauses become positive clauses under the renaming. Theory resolution then is positive hyper-resolution of goal clauses into a theory clause as the nucleus, yielding another (positive) goal clause. With any propositional symbol A in T we associate two new symbols A_+ and A_- which will be used to denote positive and negative A-literals, respectively. Also, if $\succ'$ is any extension of $\succ$, we shall order the new atoms according to $A_\sigma \succ' B_\tau$ iff $A \succ' B$, for any two atoms $A \neq B$. To represent the intended semantics of the new atoms we assume the presence of the, respectively, positive and negative connections $A_+ \vee A_-$ and $\neg A_+ \vee \neg A_-$. Let K [K_-] denote the matrix consisting of all the [negative] connections.

If C is a clause in T of the form

$$C = \sigma_1 A^1 \vee \ldots \vee \sigma_k A^k \vee \tau_1 B^1 \vee \ldots \vee \tau_m B^m$$

such that the A^i are the maximal atoms in C, that is, they are pairwise incomparable under $\succ$ and any B^j is smaller than some A^i, then by $\rho(C)$ we denote the renamed clause

$$\neg A^1_{\overline{\sigma_1}} \vee \ldots \vee \neg A^k_{\overline{\sigma_k}} \vee B^1_{\tau_1} \vee \ldots \vee B^m_{\tau_m} \ .$$

For instance, if $C = \neg A \vee \neg B \vee C$ (which can be written as $-A \vee -B \vee C$) with maximal atoms B and C we obtain $\rho(C) = \neg B_+ \vee \neg C_- \vee A_-$. Observe that for any set of clauses C_i, the entailment $C_1, \ldots, C_k \models C_0$ is valid if and only if $\rho(C_1), \ldots, \rho(C_k), K \models \rho(C_0)$. Also, for the weak extension $\succ'_w$ of $\succ'$ we observe that $\rho(C) \succ'_w \rho(D)$ if and only if $C \succ'_w D$.

With these constructions and under the assumptions made above we obtain this Lemma:

Lemma 8.4 *If I is a model of $\rho(T) \cup K_-$ then there exists an interpretation I' such that $I \subseteq I'$ which is a model of $\rho(T) \cup K$.*

Proof. Let $\succ'$ be any admissible total extension of $\succ$. We may assume that no clause in T contains more than one occurrence of its maximal (in $\succ'$) atom. (Otherwise, as T is saturated by ordered positive and negative factoring, the clause is redundant and we may remove it without affecting saturation.) We define, for any atom A in T, using induction over $\succ'$, interpretations I'_A and E_A in the following way:

$$I'_A = I \cup \bigcup_{A \succ B} E_B$$

and (i) E_A is the empty set, if either A_+ or A_- is in I, or else (ii) E_A is $\{A_-\}$ whenever $I'_A \cup \{A_-\}$ is a model of $\rho(M)$, or else (iii) E_A is $\{A_+\}$, otherwise. Finally, let

$$I'_A = I \cup \bigcup_A E_A$$

By construction I' satisfies the connections K. Assume that A is minimal such that $I'_A \cup E_A$ is not a model of $\rho(M)$. By the induction hypothesis, I'_A is a model of $\rho(M)$ and hence $E_A = \{A_+\}$. Moreover, I'_A is a model of all clauses $\rho(C)$ such that C contains only atoms smaller (in $\succ'$) than A. There exists a clause of the form $\neg A_+ \vee D_1$ in $\rho(M)$ such that D_1 is false in I'_A. There also exists a clause of the form $\neg A_- \vee C_1$ in $\rho(M)$ such that C_1 is false in I'_A. (For, otherwise, $I'_A \cup \{A_-\}$ would be a model of $\rho(M)$, hence $E_A = \{A_-\}$.) As neither C_1 nor D_1 contain A_+ they are also false in $I'_A \cup \{A_+\}$. Suppose that $C \vee A$ and $D \vee \neg A$ are clauses in T for which $\rho(C \vee A) = \neg A_- \vee C_1$ and $\rho(D \vee \neg A) = \neg A_+ \vee D_1$, respectively.

$C \vee A$ cannot be in $\mathcal{R}^{\succ_w}(T)$. For otherwise there exist clauses $D_1, \ldots, D_k$ in T such that $D_1, \ldots, D_k \models C \vee A$ and $C \vee A \succ_w D_i$, hence $C \vee A \succ'_w D_i$. Therefore, the D_i contain only atoms smaller than A in $\succ'$ so that the $\rho(D_i)$ are true in I'_A. As $\rho(D_1), \ldots, \rho(D_k), K \models C_1 \vee \neg A_-$, $C_1 \vee \neg A_-$ would have to be true in I'_A, which is a constradiction. Similarly, $D \vee \neg A$ cannot be redundant in T.

Consider the inference by ordered resolution (with respect to $\succ$) from premises $C \vee A$ and $D \vee \neg A$ resolving A. As the inference is redundant in T, either $C \vee D$ is in T or else there exist clauses $G_1, \ldots, G_k$ in T which are smaller than $D \vee \neg A$ in $\succ_w$ such that $G_1, \ldots, G_k, A \models C \vee D$. If $C \vee D$ is in T then $\rho(C \vee D)$, as it does not contain neither A_+ nor A_-, is true in I'_A. As

$\rho(C \vee D)$ and $C_1 \vee D_1$ have the same truth value in I'_A, this is a contradiction. If $C \vee D$ is not in T consider the clauses G_i. By definition of $\succ'_w$ their atoms are smaller than A with respect to $\succ'$, and hence their renamed forms $\rho(G_i)$ are true in I'_A. We may infer that $\rho(G_1), \ldots, \rho(G_k), A_+, K \models C_1 \vee D_1$. In sum, $C_1 \vee D_1$ is true in $I'_A \cup E_A$, which is a contradiction. $\square$

To finish the completeness proof for ordered theory resolution, consider any well-founded completable extension $\succ'$ of $\succ$ and ordered resolution $\mathsf{R}^{\succ'}$ with a selection function whereby all negative literals in $\rho(T)$ and K_- are selected. The given goal clauses are renamed, employing the new vocabulary, into purely positive clauses. The positive connections are not considered, at first. Then the only non-redundant inferences apart from factoring are ordered hyper-resolution inferences with non-goal clauses as electrons into a nucleus that is either a renamed theory clause or else one of the connections in K_-. The former kind of inferences correspond to theory resolution, while the latter represent ordered resolution between two non-goal clauses. If no contradiction can be derived, the union of the renamed goal clauses with $\rho(T)$ and K_- are satisfiable. If I is a model of this clause set, by the Lemma it can be extended to a model of K and $\rho(T)$. As the renamed goal clauses are positive, and inferences create positive clauses only, the extension of the model also satisfies the latter.

The redundancy criterion for the theory resolution inference is simply the redundancy criterion for general ordered resolution instantiated for the corresponding hyper-resolution inference between the renamed goal clauses and the renamed theory clause. In particular, certain clauses which denote tautologies before transformation, are not redundant. This is similar to what we have seen in lock resolution.

If T is saturated with respect to a total ordering $\succ$, there is no difference between an ordered theory resolution inference and an ordered resolution inference in which one premise is a goal clause and the second premise is a theory clause. In this case no extension and renaming is needed to achieve the same effect. With a partial ordering the residuums in a theory resolution inference may be shorter. On the other hand there are more theories that can effectively (and finitely) be saturated under a total atom ordering. For an example see Section 10.4

9 First-Order Resolution Methods

9.1 First-Order Sequents

In this paper, resolution has been described for (possibly infinite matrices of) ground formulas. In practice, resolution methods for propositional logic only play a minor role compared to the Davis-Putnam procedure (that we briefly deal with in Section 10.1) or the method of Ordered Binary Decision

Diagrams (Bryant 1992). One of the main applications of resolution and related saturation-based methods is automated theorem proving for first-order logic. In this context one is interested in finding the proof of a given sequent $N \vdash M$, possibly containing variables and quantifiers. To be able to apply resolution, quantifiers have to be eliminated, and the sequent has to be replaced by the equivalent negative matrix $N, \neg M \vdash \bot$. Inference systems and redundancy criteria have to be lifted to (quantifier-free) clauses with variables.

9.2 Lifting of Resolution Inferences

The lifting of unconstrained resolution inferences to quantifier-free clauses with variables is not a problem and is dealt with by standard methods. In particular equality of ground atoms is generalized to unification of non-ground atoms. For instance, general (ground) resolution

$$\frac{F[G] \quad F'[G]}{F[G/\bot] \vee F'[G/\top]}$$

is lifted to

$$\frac{F[G_1, \ldots, G_k] \quad F'[G'_1, \ldots, G'_n]}{F\sigma[G/\bot] \vee F'\sigma[G/\top]}$$

where σ is the *most general unifier* (mgu) of the atoms $G_1, \ldots, G_k$, $G'_1, \ldots, G'_n$, $G = G_1\sigma$, and where one implicitly renames variables in one of the premises in order to achieve disjointness of variables. This formulation takes care of the fact that in the ground inference all occurences of G are replaced both in F and F'. On the non-ground level the set of positions at which G occurs has to be determined (non-deterministically) by unification. This is similar for all inferences in which more than one occurence of an atom is to be replaced. An additional dimension of non-determinism is caused by the fact that there may not be a unique choice of atoms G_i and G'_j that can be unified. We refer to the discussion about replacement strategies in Section 6.2.

9.3 Lifting of Ordering Constraints

Theorem proving processes are parameterized by orderings, selection functions, renaming strategies, simplification and deletion strategies. Here is where lifting starts to become less straightforward. In the literature we find two main techniques for lifting ordering constraints. A possible choice is to approximate ground constraints safely on the non-ground level. One first extends the (total, wellfounded) ground ordering to (a partial, wellfounded ordering on) non-ground expressions E and E' by stipulating that $E \succ E'$ if and only if $E\sigma \succ E'\sigma$, for all ground substitutions σ. Then one lifts ordering ground constraints of the form $E \succ E'$ into constraints of the form $E' \not\succeq E$ on the non-ground level. For instance (simple, binary) ordered resolution on the non-ground level becomes the inferences

$$\frac{C \vee A \quad D \vee \neg B}{C\sigma \vee D\sigma}$$

where σ is the most general unifier of A and B such that (i) $A'\sigma \not\succeq A\sigma$, for any A' in C, (ii) C contains no selected literal, (iii) either $\neg B$ is selected or else $B'\sigma \not\succ B\sigma$, for any atom B' in D. Note that it is better to check the constraints after the unifier has been applied as this gives a more precise approximation of the ordering constraints for the ground instances of the inference. For many orderings, the satisfiability of non-ground constraints will be undecidable. In such cases one has to employ safe approximations, that is, possibly incomplete, but in any case sound, constraint solvers.

9.4 Constrained Formulas

A second, perhaps more adequate method is based on considering non-ground expressions with constraints. The constraints restrict the set of ground terms that one may substitute for a variable. Notations such as

$$C\ [\gamma]$$

are used to denote the set of all ground instances of a non-ground clause C such that the constraint γ is satisfied. (Binary) ordered resolution (without selection) on clauses can be lifted into the inference

$$\frac{C \vee A\ [\gamma] \quad D \vee \neg B\ [\delta]}{C \vee D\ [\gamma \wedge \delta \wedge (A > C) \wedge (B \geq D) \wedge (A \approx B)]}$$

on constrained non-ground clauses. The resolvent inherits the constraints γ and δ of the premises to which the maximality contraints $(A > C) \wedge (B \geq D)$ and the equality constraint $A \approx B$ for the resolved atoms are added. Such notations have been inspired by constraint logic programming and, in the context of automated theorem proving, been suggested by Kirchner, Kirchner & Rusinowitch (1990), among others. Completeness proofs for certain saturation-based theorem proving strategies involving constrained clauses have first been obtained by Huet (1972), Bürckert (1990) and Nieuwenhuis & Rubio (1992). By moving the constraints from the meta-level to the object level, and through constraint inheritance, the ordering restrictions of the ground level are represented in a precise way and no information is lost a priori. The significance of constraint notations also becomes apparent when finite representations of saturated theories are sought, cf. Section 10.4.3.

Constraints of the above form combine logical and meta-logical restrictions into one expression. In the binary resolution inference, $A \approx B$ is the logical constraint that ensures soundness of the inference. There is some evidence for the fact that notations which explicitly separate logical from meta-logical constraints would be more appropriate. A meta-logical constraint can be relaxed without affecting soundness and completeness, and it is the case that certain simplification strategies actually require such constraint relaxation. Also, solvers for meta-logical constraints need not be complete. Of course, logical constraints must not be relaxed, and completeness of the theorem proving process requires that one is able to decide their solvability.

9.5 Resolution Modulo an Equational Theory

The concept of constraints provides a framework in which the results about resolution can be easily extended to resolution *modulo equational theories* $\approx$ (on ground atoms). On the ground level one would only consider formulas that contain atoms which are canonical representations of their equivalence classes. Ground resolution inferences do not introduce any non-canonical atoms if the atoms in the premises are canonical. The atom ordering $\succ$ needs to be compatible with $\approx$. As the only requirement is well-foundedness of $\succ$ on the canonical atoms, for any equivalence relation such orderings can be found. For attaining refutational completeness only solvability of equality constraints needs to be decidable. In case where the sets of unifiers can be very large (e.g., in AC-unification) or even infinite (e.g., in the case of higher-order unification) constraints are an indispensable concept (Huet 1972, Nieuwenhuis & Rubio 1994, Vigneron 1994).

9.6 Redundancy

Most simplification and deletion techniques are easily extended to the non-ground case on the basis of this straightforward lifting of redundancy: A non-ground clause (matrix, sequent, inference) is called *redundant* in Σ if and only if all its ground instances are redundant in Σ.

Some cases of subsumption, however, are not covered by this method of lifting redundancy criteria. For instance the non-ground atom $p(x)$ properly subsumes the atom $p(f(y))$, hence, in the presence of the former, one wants to delete the latter. Such a deletion rule is not directly justified by the non-ground form of redundancy as the respective ground instances are identical. On the other hand it is intuitively clear that it suffices to only compute with one non-ground representation of any ground formula. For covering all cases of subsumption by the standard redundancy criterion, one needs to add further syntactic notation. One possiblity is to consider ground instances $C\sigma$ of a non-ground clause C as *labeled clauses* of the form $C : C\sigma$

carrying the clause C from which the instance is obtained as a label. In that case clause orderings can be conceived that lexicographically combine any admissible ordering on unlabeled clauses in the sense of this paper with an arbitrary well-founded ordering such as the subsumption ordering for non-ground clauses.

To formalize all aspects of lifting properly and independently of concrete notations for non-ground clauses (with or without constraints, with or without typings for variables, with or without an equational background theory) one should employ appropriate abstract notions of *representation* and *approximation* that relate a non-ground expression to the set of ground formulas it represents. Such an approach would be related in spirit to the notion of abstract interpretation in dataflow analysis frameworks.

10 Applications

We briefly describe some areas to which this theory of resolution can be fruitfully applied. To keep this paper within reasonable bounds our exposition will have to remain sketchy.

10.1 The Davis-Putnam Method

A very effective method for testing satisfiability of a finite set of propositional standard clauses is the Davis-Putnam method (Davis & Putnam 1960). The method consists in a combination of eagerly perfomed simplification steps (unit reduction, pure literal detection) together with case splits on propositional variables. We can model this method and justify its completeness in our framework if we extend or notion of linear theorem proving derivations on matrices to tree-structured derivations as created by derivation steps of this form:

Splitting

$$N \;\rhd\; N, M_1 \quad | \quad \ldots \quad | \quad N, M_k \qquad (k \geq 1)$$

if N is satisfiable if and only if one of the N, M_i is satisfiable

Splitting branches the theorem proving process into k subprocesses with initial matrices N, M_i. That is, instead of linear derivation sequences one now consideres possibly infinite, finitely branching trees in which each node represents one of the derivation steps deduction, deletion or splitting. In tree derivations, inference system and redundancy criteria need not be uniform throughout the tree. Therefore, suppose we have families $\mathcal{J} = (\mathcal{J}^\nu)_\nu$ and $\mathcal{J}_0 = (\mathcal{J}_0^\nu)_\nu$ of inference systems such that $\mathcal{J}_0^\nu \subseteq \mathcal{J}^\nu$, for every ν, and having $\mathcal{R} = (\mathcal{R}^\nu)_\nu$ as associated family of redundancy criteria. Assume also that every system $\mathcal{J}_0^\nu$, together with $\mathcal{R}^\nu$, is refutationally complete. The theorem proving derivation tree (based on the families $\mathcal{J}$ and $\mathcal{R}$) is called *fair* with respect to $\mathcal{J}_0$ if, for the limit $N_\infty = \bigcup_i \bigcap_{j \geq i} N_j$ of each path $N_0, N_1, N_2, \ldots$ in the tree, there exists a ν such that every inference in $\mathcal{J}_0^\nu$ with non-redundant premises in N_∞ is redundant in $\bigcup_j N_j$ with respect to $\mathcal{R}^\nu$, i.e., $\mathcal{J}_0^\nu(N_\infty \setminus \mathcal{R}^\nu(N_\infty)) \subseteq \mathcal{R}^\nu_{\mathcal{J}_0^\nu}(\bigcup_j N_j)$. As in Lemma 4.4 we can show that for a fair derivation the limit of any path is saturated up to redundancy. In that case the initial matrix N_0 is unsatisfiable if and only for every path $N_0, N_1, N_2, \ldots$ its limit $\bigcup_i \bigcap_{j \geq i} N_j$ is unsatisfiable, that is, $\bigcup_j N_j$ contains a contradiction.

The following rules which describe the Davis-Putnam method are specific instances of deduction, deletion and splitting.

Unit reduction

$N, L, C \vee \overline{L} \ \rhd \ N, L, C$ where $\overline{L}$ is the complement of L

Unit subsumption

$N, L, C \vee L \ \rhd \ N, L$

Pure literal extension

$N \ \rhd \ N, L$ if L, but not its complement, occurs in N

Tautology deletion

$N, C \ \rhd \ N$ if C is a tautology

Splitting

$N \ \rhd \ N, A \ \mid \ N, \neg A$ if the atom A occurs in N

The strategy is such that the splitting rule is applied only if no other rule is applicable. Pure literal extension is also an instance of splitting (for $k = 1$). If L, but not its complement, occurs in N one only needs to consider interpretations in which L is true. If N is satisfiable then it also has a model in which L is true. Pure literal extension triggers subsequent subsumption steps by which all clauses that contain that literal can be eliminated.

It is easy to see that the Davis-Putnam method, when applied to matrices that consist of standard ground clauses, represents a fair resolution strategy. In fact, the only case in which no rule is applicable is when either N contains the empty clause, or else N contains only literals and no two of them are complementary. For such matrices, no resolution and/or factoring inferences are applicable, hence they are saturated with respect to standard resolution $\mathsf{R}_S^\succ$ and an arbitrary atom ordering $\succ$. The completeness of the method is a consequence of Theorem 7.1. It is also easy to see that the method terminates whenever the initial matrix is a finite set of standard ground clauses. The main problem with respect to performance in practice is which atom A to select in the splitting step. The method can be further improved by adding additional simplification and deletion steps such as general subsumption and subsumption resolution (also see Section 6.5):

Subsumption resolution

$N, D \vee L, D \vee C \vee \overline{L} \ \triangleright\ N, D \vee L, D \vee C$

if $\overline{L}$ is the complement of L.

Note that the main technique in the Davis-Putnam procedure, splitting on an atom A, followed by unit reduction of the two branches by A or $\neg A$, respectively, is an instance of ordered self-resolution:

Splitting as self-resolution

$$\frac{M, F[A]}{M, F[A/\bot], F[A/\top]}$$

In that view, the derivation tree is represented as a single clause M, F in which the component formulas F represent the leaves of some partially expanded Davis-Putnam tree. Each formula itself is in conjunctive normal form. The self-resolution inference takes one leaf F in the tree and self-resolves in F on the maximal atom A, where the ordering is implicitly determined by the process. Replacement of A by $\bot$ and $\top$ is unit reduction. If A occurs in just one polarity in F, the conclusion simplifies to either $M, F[A/\bot]$ or $M, F[A/\top]$.

Despite this fact it does not seem to be possible to model the Davis-Putnam procedure as a *linear* theorem proving process based on ordered resolution with selection in a natural way. The choice of atoms that are self-resolved at any step in the Davis-Putnam tree does not follow one uniform atom ordering but is guided by the syntactic specificities of the respective subtrees. That is, in different subtrees splitting may occur in different order. Any linear ordered resolution process would have to adopt a uniform ordering.

On the other hand one can show that in a theorem proving derivation one may, without affecting the general results about how to effectively achieve fairness and, hence, the saturation of its limit, always admit *finitely* many "heureka" steps $N \triangleright N'$ in which one replaces N by any N' such that both consistency and inconsistency is preserved. All this indicates why the Davis-Putnam procedure does not easily extend to the infinite case of clauses with variables.

10.2 Saturated Semantic Tableaux

While the Davis-Putnam method splits on tautologies of the form $A \vee \neg A$, the tableau method performs case splits according to the clauses in a matrix. In the most basic form, in a propositional and clausal setting, we may describe it by the following two forms of derivation steps:

Splitting on clauses

$$N, (L_1 \vee \ldots \vee L_k) \ \triangleright\ N, L_1 \quad | \quad \ldots \quad | \quad N, L_k$$

Ancestor literal complement

$$N, L, \overline{L} \ \triangleright\ \bot$$

Splitting on clauses is a combination of splitting and subsumption.

$$N, C \ \triangleright\ N, C, L_1 \quad | \quad \ldots \quad | \quad N, C, L_k \ \triangleright\ N, L_1 \quad | \quad \ldots \quad | \quad N, L_k$$

if $C = L_1 \vee \ldots \vee L_k$. Since we eliminate the clause on which the case split is performed, the tablaux become automatically regular (no path contains two nodes that correspond to the same clause split). Closing a path in the presence of a complementary ancestor literal is a unit resolution step, generating the empty clause, followed by subsumption steps by which all other clauses are eliminated. A matrix to which none of the above rules is applicable consists either of the empty clause, or else of unit clauses such that no two of them are complementary. Any strategy that applies the two rules exhaustively (and don't-care nondeterministically) is fair, hence the limit of each path in the derivation tree is closed under resolution. The completeness of the tableau method thus follows from the completeness of standard resolution (cf. Theorem 7.1). In the basic formulation orderings are irrelevant.

We may improve the method by adding ordering constraints, selection functions and simplification. It is sufficient to saturate the limit of each path with respect to any of the resolution calculi. To that end we may restrict clause splitting to instances that correspond to a step of ordered resolution with selection.

Splitting on clauses induced by ordered resolution

$$N, C \vee A \vee \ldots \vee A, D \vee \neg A \ \triangleright\ N, C, \neg A \quad | \quad N, A, D$$

if the resolution inference

$$\frac{C \vee A \vee \ldots \vee A \quad D \vee \neg A}{C \vee D}$$

is an inference by standard ordered resolution with respect to a given ordering and selection function.

In this formulation, rather than splitting $\neg A \vee D$, with $D = L_1 \vee \ldots \vee L_k$, into its $k+1$ disjuncts, we split into two cases where the first case corresponds to the atom on which we have resolved, and where the second case is the remaining disjunct D which may be split further in subsequent steps. We have exploited the fact that in the second of the two cases we may assume that A is true, and hence have simplified $C \vee A$ into A. This formulation of the semantic tableau emphasizes its close relationship to the Davis-Putnam methods. The difference between the two methods seems to be mainly due to the choice of concrete datastructures. In semantic tableaux when performing an expansion of the form

$$N, L_1 \vee \ldots \vee L_k \ \rhd \ N, L_1 \quad | \quad \ldots \quad | \quad N, L_k$$

the common part N is not duplicated but shared in the graphical representation of the tableaux. In fact one may even consider the initial clause set N_0 as implicit and only store the literals L_i on which case splits are performed in the tableau. But then properties such as regularity have to be posed as extra constraints. When one translates the above formulation of a semantic tableau with ordering restrictions and selection functions into the more common graphical formulations one obtains in particular a justification of the ordering restrictions for tableaux that have been proved complete by Klingenbeck & Hähnle (1994). Actually, one can also see that the ordering $\succ$ does not need to be uniform. For each path one may choose a specific ordering.

In short, what we are saying is that with the theoretical machinery of this paper one may generalize the notion of a closed tableau to that of a *saturated tableau*. A saturated tableau is one in which all paths are saturated up to redundancy with respect to one of the refutationally complete calculi of resolution.

10.3 Model Elimination

Consider the method of semantic tableau applied to standard matrices of the form T, G, where G is a positive clause $A_1 \vee \ldots \vee A_k$ and where every clause in T has a negative literal. Assume a selection function which selects exactly one negative literal in any clause in T. This is the format that we can obtain after renaming whenever T is a consistent set of clauses. In fact, whenever some matrix N is inconsistent, we may find an inconsistent finite submatrix of the form T', G' where G' is a clause such that T' is consistent but T', G' is not. After renaming, cf. the discussion of semantic and set-of-support resolution in Section 8.1, we obtain T, G as assumed.

Model elimination (Loveland 1969) can be described by this calculus of theorem proving derivation rules:

Splitting on the goal clause

$$T, G \ \rhd \ T, A_1 \quad | \quad \ldots \quad | \quad T, A_k$$

Expansion with a theory clause

$$T_1, (\neg A \vee L_1 \vee \ldots \vee L_n), M, A \rhd \ T_1, M, A, L_1 \quad | \quad \ldots \quad | \quad T_1, M, A, L_n$$

if $\neg A$ is selected in the indicated theory clause

Ancestor literal complement

$N, L, \overline{L} \;\triangleright\; \bot$

The expansion rule is a special case of the rule "splitting on clauses" of Section 10.2 for a theory clause, where the branch for $\neg A$ has been omited since it immediately derives $\bot$ from A and $\neg A$. Again we delete the clause on which the splitting is performed, avoiding a multiple case analysis over the same clause. The submatrices M contain newly derived literals from either the initial goal splitting or any subsequent expansion step.

Model elimination refines semantic tableau in that an expansion step by a theory clause must be triggered by the existence of a literal that descends from the initial goal clause G and that is complementary to the literal that is selected in the theory clause. The expansion is therefore linear with respect to the theory clauses. Clearly, in any fully expanded model elimination tree, any path is saturated under resolution with selection up to redundancy. Therefore, T, G is inconsistent if and only if any path in the tree ends with $\bot$. Note that since in concrete applications we do not effectively know what the renaming of T', G' into T, G looks like, we have to compute with the original T', G' (we may even have to consider more than one tree, one for every clause G' in $N \setminus T'$). In particular we do not know which ones the selected literals are, so that we have to try them all in a non-deterministic fashion.

We may now come back to a question about free selection that was raised earlier in Section 8.2. To that end we analyse the correspondence between prefixes in model elimination trees and resolution derivations. In a model elimination tree, any inner node contains a unique literal that has been introduced in the splitting step which has produced that node (these are the A_i in the goal splitting step, and the L_j in the expansion steps). Let us call this literal the *main literal* of the node. Now take any (partial) derivation tree that contains at least one expansion step and for which none of the leaf nodes contain a pair of contradictory literals. Let $L_1, \ldots, L_m$ be the main literals of the sequence of leafs (ignoring any $\bot$-leaves) in that prefix. Then the clause $L_1 \vee \ldots \vee L_m$ can be derived from T, G by linear resolution (at most one premise is a clause in T) together with implicit or explicit factoring. This gives the linearity constraints in SL-resolution (Kowalski & Kuehner 1971). The selection constraints for that method allow to select an arbitrary literal that was inherited by a resolvent from the theory clause premise of the resolution step. That selection simply means that the model elimination tree can be expanded don't-care non-deterministically by selecting any of its leafs. The ancestor literal complement steps in model elimination either correspond to implicit factoring, or else to subsumption resolution steps called "ancestor resolution" in (Kowalski & Kuehner 1971). Tautology elimination as well as the other redundancy elimination techniques of that paper can be easily modeled by standard redundancy.

In conclusion, there is a close correspondence between model elimination trees and certain resolution derivations with simplification. Resolution methods such as SL-resolution which exploit that relationship, may feature combinations of restrictions on the matrix level that can be justified by the semantic framework with restrictions on the level of theorem proving derivation trees that can be justified by proof transformations.

10.4 Effective Saturation of First-Order Theories

Saturation up to redundancy terminates in many cases of consistent theories, if strong enough techniques for simplification and redundancy elimination are employed. In this section we briefly describe some of the applications in which finitely saturated theories play a central role. We intend to demonstrate that saturation can be understood as a (partial) compilation process through which, when it terminates, certain objectives with regard to efficiency can be achieved in an automated way.

10.4.1 Decision Procedures Based on Resolution

The abstract notion of redundancy is sufficiently general so as to accomodate virtually all the major techniques of simplification and elimination. With the right setting of the resolution parameters, most of the known decidable fragments of first-order logic can be decided by saturation up to redundancy. The theory of resolution is, hence, a powerful tool for obtaining proofs of decidability for first-order theories and for logics that can be semantically embedded into first-order logic. An early example was given by Joyner Jr. (1976) where he shows that the monadic class can be decided by ordered resolution, endowed with subsumption and condensement as simpification techniques. Since then, many other decidable classes have been shown decidable by suitably refined calculi of ordered resolution (and paramodulation): the monadic class with equality (Bachmair, Ganzinger & Waldmann 1993), the Ackermann class with equality (Fermüller & Salzer 1993), a subclass of Maslov's class K (Fermüller, Leitsch, Tammet & Zamov 1993) and various logics for knowledge representation (Fermüller et al. 1993, Hustadt 1997). Hustadt (1997) appears to be the first to describe a resolution-based decision procedure for the full Maslov class K, the completeness proof of which makes essential use of the methods presented here, and in particular of renaming techniques. Schmidt (1997) proposes a general method for obtaining resolution-based decision procedures for many modal logics. In particular she provides a descision procedure for an interesting fragment of first-order logic, called *path logic*. Fermüller et al. (1993) give a comprehensive overview of the earlier work on decision methods based on resolution.

10.4.2 Automated Complexity Analysis

Motivated by work on "local theories" by McAllester (1993), the relation between saturation and decidability issues has been extended to complexity analysis by Basin & Ganzinger (1996) . There is was shown that the complexity of the ordering that one uses for saturation is directly related to the complexity of the entailment problem for a theory. More precisely, suppose N is a standard matrix (with variables). The entailment problem for the *theory* N consists in checking whether or not a *query* C, where C is a ground standard clause, is logically implied by N. If N is saturated (up to redundancy) under standard ordered resolution without selection (that is, no atom is selected in any clause and, hence, the resolved atom is maximal in both premises), one can derive upper bounds for the complexity of the entailment problem for N. For this to be possible, the ordering on atoms must be such that for any given ground atom A there are are only finitely many ground atoms that are smaller than A. In that case, the *complexity* of an ordering can be bounded from above by a function f such that whenever A is a ground atom of size (number of symbols) n, then there are at most $f(n)$ ground atom that are smaller than n. One of the main results in (Basin & Ganzinger 1996) is the following:

Theorem 10.1 *Suppose $\succ$ is a partial well-founded ordering on ground atoms of complexity f and N is a finite set of* Horn *clauses that is saturated under ordered resolution up to redundancy with respect to each total, well-founded extension of $\succ$. Then the entailment problem for N is decidable in time $O(f^k)$ where k is a constant that depends only on the theory N.*

In particular, the entailment problem is polynomial, if the ordering is of polynomial complexity.

For the proof idea (assume, for simplicity, $\succ$ is total) note that if N is saturated, all inferences in which both premises are in N are redundant. To decide as to whether some ground query C is entailed, negate C, add the resulting unit clauses to N, and restart saturation by ordered resolution. Clauses generated from ordered inferences with one premise in N and one premise in $\neg C$ cannot generate (ground instances of) clauses in which an atom is bigger than each atom in C. Hence, if C is a consequence of N, it already follows from a set of ground instances of N in which all atoms are smaller or equal in $\succ$ to an atom in C. By applying dynamic programming methods of bottom-up computation (N was assumed to be Horn), the result follows.

What we have just described, can be summarized into this Lemma:

Lemma 10.2 *Let N be saturated up to redundancy with respect to ordered resolution (without selection) based on a total and well-founded ordering $\succ$.*

If C is a standard ground clause then C is a logical consequence of N if and only if C is a logical consequence of those ground instances D of N in which for any atom A in D there exists an atom B in C such that $B \succeq A$.

It is sometimes the case, for examples see (Basin & Ganzinger 1996), that the natural presentation of a theory is not saturated with respect to a desired ordering, but can be finitely saturated. In such cases saturation can be viewed as an optimizing compiler which adds sufficienty many "useful" consequences to a theory presentation so as to achieve a certain complexity bound for its entailment problem.

10.4.3 Deduction with Saturated Presentations

Having discussed why lifting methods in practice may require constraint notation, we will give an example in which such notation allows to finitely present a saturated theory such that theory resolution gives a practically useful refinement of resolution modulo that theory.

The following is a presentation T, using ordering-constrained clauses, of the transitive-reflexive closure p^* of a binary predicate p:

$$
\begin{array}{rcll}
 & \rightarrow & p^*(x,x) & & (1) \\
p(x,y) & \rightarrow & p^*(x,y) & & (2) \\
p^*(x,y),\, p^*(y,z) & \rightarrow & p^*(x,z) & & (3) \\
p^*(x,y),\, p(y,z) & \rightarrow & p^*(x,z) & [y \succ z, y \succ x] & (4) \\
p(x,y),\, p^*(y,z) & \rightarrow & p^*(x,z) & [x \succ y, x \succ z] & (5) \\
p^*(x,y),\, p(y,z) & \rightarrow & p^*(x,z) & [z \succ x, z \succ y] & (6) \\
p(x,y),\, p(y,z) & \rightarrow & p(x,z) & [y \succ x, y \succ z] & (7)
\end{array}
$$

The presentation is saturated (under ordered resolution with selection) with respect to a specific (class of) orderings. Suppose we are given any well-founded and total ordering on ground terms. Then the ordering on atoms has to be such that (i) $A \succ B$ whenever the maximal term in A is greater then the maximal term in B, and (ii) $p^*(s,t) \succ p(u,v)$, whenever the maximal term among s and t is the same as the maximal term among u and v. The selection function always selects the maximal negative atom, provided the maximal term of the clause occurs in a negative p^*-atom, or else in a negative p-atom but not in a positive p^*-atom. In all other cases, nothing is selected. To check that the system is in fact saturated is not difficult, but somewhat tedious. With a sufficiently powerful saturation procedure such as the one implemented in the Saturate system (Ganzinger & Nieuwenhuis 1994) it can be obtained automatically from the first three clauses.[9] For instance

[9] Actually in the system that we get automatically from (1)–(3), instead of (7), which is not a consequence of (1)–(3), we find the clause $p(x,y),\, p(y,z) \rightarrow p^*(x,z)$ $[y \succ x, y \succ z]$. Since in the application we are interested in one may assume non-theory clauses to not contain negative occurrences of p its replacement by (7) is justified.

consider the inference

$$\frac{p(x,y),\, p^*(y,z) \to p^*(x,z)\ [x \succ y, x \succ z] \quad p^*(x,z),\, p^*(z,u) \to p^*(x,u)}{p(x,y),\, p^*(y,z),\, p^*(z,u) \to p^*(x,u)\ [x \succ y, x \succ z, x \succeq u]}$$

by ordered resolution with selection from the fifth and the third clause. For $p^*(x,z)$ to be selected in the second premise, $x \succeq u$ must be true and, hence, has been added to the constraint of the conclusion. The following proof (for the case $x \succ u$, the case $x = u$ being trivial) of the same clause involves only clauses smaller than the second premise:

$$\dfrac{p(x,y) \qquad \dfrac{p^*(y,z),\, p^*(z,u)}{p^*(y,u)}\ [x \succ y, x \succ z, x \succ u]}{p^*(x,u)}\ [x \succ y, x \succ u]$$

The constraints indicate that the involved clauses do exist and are sufficiently small.

Given that T is saturated, consider saturation of matrices T, N that contain T as a subtheory. We may assume that p-atoms occur in N only positively, while p^*-atoms occur only negatively in N. To put it differently, any negative [positive] occurrence of p [p^*] in N can be replaced by p^* [p] without affecting satisfiability or unsatisfiability of T, N. Consider ordered inferences from mixed premises in T and N:

(a) Inferences with premise (3) or (4) and any clause in N are impossible: In (3) and (4) a p^*-atom is selected while N has no positive occurrences of p^*.

(b) Omitting the negative premise, hyper-resolution inferences between clauses in N as positive premises and clause (7) as negative premise can be written as this inference (we describe the ground version):

Ordered chaining with selection, right

$$\frac{C, p(s,t) \quad D, p(t,u)}{C, D, p(s,u)}$$

> where (i) $t \succ s$, $t \succ u$, (ii) $p(s,t)$ is greater than any atom in C, (iii) $p(t,u)$ is greater than any atom in D, and (iv) neither C nor D contain a selected atom.

(c) In the clauses (5) and (6) the positive atom is maximal. Ordered resolution with positive premise (5) or (6) and a N-clause as negative premise, followed by the resolution of the resulting negative p-atom by another clause in N with a positve p is tantamount to the two cases of ordered chaining left:

Ordered chaining with selection, left (I)

$$\frac{C, p(s,t) \quad D, \neg p^*(s,u)}{C, D, \neg p^*(t,u)}$$

where (i) $s \succ t$, $s \succ u$, (ii) $p(s,t)$ is greater than any atom in C, (iii) $p^*(s,u)$ is either selected or else greater than or equal to any atom in D, and (iv) C contains no selected atom.

Ordered chaining with selection, left (II)

$$\frac{C, p(t,s) \quad D, \neg p^*(u,s)}{C, D, \neg p^*(u,t)}$$

where (i) $s \succ t$, $s \succ u$, (ii) $p(t,s)$ is greater than any atom in C, (iii) $p^*(u,s)$ is either selected or else greater than or equal to any atom in D, and (iv) C contains no selected atom.

Inferences involving (1) or (3) need not be dealt with specifically. Note that the chaining inferences preserve the specific normal form with respect to the polarity of occurrences of p and p^*.

In summary, when one extends standard ordered resolution with selection by the ordered chaining inferences one obtains a refutationally complete specialization of resolution for theories with transitive and reflexive relations p^* in which no resolution inferences with the transitvity axiom for p^* are required.[10] In Bachmair & Ganzinger (1994*b*) we have employed specific methods from term rewriting to obtain a closely related result. Here we have demonstrated that processing a theory presentation by saturation, which can be automated to a large extent, can mechanically produce a practically useful inference system that might otherwise require non-trivial meta-level arguments for its completeness proof.

It seems that many other theories, including congruence, orderings, and distributive lattices, can be engineered in a similar way. This sheds some new light on how theory resolution can actually be efficiently implemented in practice for theories that can effectively be saturated in a way such that "problematic" clauses such as transitivity are eliminated to a certain extent. Pre-saturation of theory modules contributes to an efficient handling of large, but structured theories. Similar ideas in the context of theorem proving for modal logics where specific saturations of the respective "background theories" form the heart of the method have been elaborated by Nonnengart (1995).

[10] The chaining inferences encode certain ordered inferences with transitivity. The advantage over unordered hyper-resolution strategies with the transitivity clause is a better balancing between forward and backward computation.

11 Concluding Remarks

We have presented a version of resolution for general clauses with ordering constraints and selection functions. Orderings of clauses are based on well-founded, partial orderings on atoms. Slagle (1967) attributes the original idea of ordering atoms to Reynolds (1965). Selection functions select atoms that must be true in interpretations in which the clause is false. The earliest resolution strategy that exploits selection appears to be hyper-resolution (Robinson 1965*a*). The don't-care non-deterministic aspects of selection in resolution and the resulting pruning of resolution search spaces have been first recognized by Kowalski & Kuehner (1971). Our main theoretical result is the refutational completeness of this family of calculi in the presence of a certain redundancy criterion based on a well-founded ordering on formulas. The proof applies a variant of the model construction technique that was originally introduced in (Bachmair & Ganzinger 1990). Related ideas had been described earlier, but not worked out with the required mathematical rigour, by Zhang (1988), and subsequently also by Pais & Peterson (1991). From that construction one observes that (in the case of standard clauses) the only inferences that are needed arise from productive clauses and from minimal counterexamples to the model property. The standard concept of redundancy is an abstraction from these restrictions that enjoys stability under deduction as well as under deletion of redundant clauses. In some cases, however, the abstraction is too crude. Recently we have shown that one can go one step further and try to use more semantic properties of these partial interpretation for further pruning the search space (Ganzinger, Meyer & Weidenbach 1997).

The concept of redundancy allows us to handle a variety of simplification techniques, such as tautology deletion, subsumption, and simplification by rewriting; and provides a framework for effectively handling equivalences. The well-foundedness assumption for atom orderings is not an essential restriction. If one assumes well-foundedness of the atom ordering, compactness of first-order logic is a consequence of our completeness result. Alternatively one might start out by assuming compactness. If a matrix is inconsistent, a finite submatrix will already be inconsistent. Restricted to that submatrix, any ordering will be wellfounded.

Resolution for standard clauses emerges as a special case in which the simplification rules for clausal normalization are applied before any resolution inferences. In particular, our completeness results applies to sharpened versions of ordered resolution and positive resolution. The same techniques can also be applied to hyper-resolution, semantic resolution, and set-of-support resolution, to the inverse method, to Boolean ring-based methods, as well as to constrained resolution and theory resolution. We have thus generalized these various completeness results in at least two ways. First, we consider arbitrary formulas, not just clauses; and secondly, we establish completeness in the presence of redundancy, which also allows us to deal with a wide spectrum of simplification mechanisms. For instance, the clausal normalization rules can be applied selectively to certain parts of formulas only, a tactics that could be useful for dealing with equivalences $F \equiv G$.

The theory has been developed mainly to serve as a theoretical justification for many techniques that have been proposed for saturation-based theorem proving. Despite the powerful criteria for redundancy that we have introduced, saturation-based theorem proving often turns out to be too little goal-oriented in practice. One should attempt to combine it with the goal-oriented methods of the sequent calculus or the semantic tableau. We refer to (Avron 1993) for some inital discussion of this problem. Having demonstrated that our theory of resolution may be useful for explaining essential properties of semantic tableau and variants thereof, including the Davis-Putnam method, model elimination and SL-resolution, is reassuring and may serve as a starting point for further investigations of this problem.

We have also indicated that there might be a smooth way of specializing resolution to theories that include transitive relations (e.g., orderings, equality) in a way such that rewriting-based calculi (e.g., ordered chaining, superposition; Bachmair & Ganzinger 1990) are obtained. This is another topic that deserves more attention with regard to future work.

References

Andrews, P. B. (1981), 'Theorem proving via general matings', *Journal of the ACM* **28**, 193–214.

Avron, A. (1993), 'Gentzen-type systems, resolution and tableaux', *Journal of Automated Reasoning* **10**(2), 265–281.

Bachmair, L. (1989), Proof normalization for resolution and paramodulation, *in* 'Proc. 3rd Int Conf. Rewriting Techniques and Applications', Vol. 355 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, pp. 15–28.

Bachmair, L. & Dershowitz, N. (1987), Critical pair criteria for rewriting modulo a congruence, *in* J. Davenport, ed., 'Proc. EUROCAL 87', Vol. 378 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, pp. 452–453.

Bachmair, L. & Ganzinger, H. (1990), On restrictions of ordered paramodulation with simplification, *in* 'Proc. 10th Conf. on Automated Deduction (Kaiserslautern, Germany), LNAI 449', Springer-Verlag, Berlin, pp. 427–441.

Bachmair, L. & Ganzinger, H. (1994*a*), 'Rewrite-based equational theorem proving with selection and simplification', *Journal of Logic and Computation* **4**(3), 217–247. Revised version of Technical Report MPI-I-91-208, 1991.

Bachmair, L. & Ganzinger, H. (1994*b*), Rewrite techniques for transitive relations, *in* 'Proc. 9th IEEE Symposium on Logic in Computer Science', IEEE Computer Society Press, pp. 384–393.
*http://www.mpi-sb.mpg.de/~hg/pca.html#LICS94

Bachmair, L., Ganzinger, H. & Waldmann, U. (1993), Superposition with simplification as a decision procedure for the monadic class with equality, *in* G. Gottlob, A. Leitsch & D. Mundici, eds, 'Proc. of Third Kurt Gödel Colloquium, KGC'93', Vol. 713 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, pp. 83–96. Revised version of Technical Report MPI-I-93-204.

Basin, D. & Ganzinger, H. (1996), Complexity analysis based on ordered resolution, *in* 'Proc. 11th IEEE Symposium on Logic in Computer Science', IEEE Computer Society Press, pp. 456–465.
*http://www.mpi-sb.mpg.de/~hg/pca.html#LICS96

Baumgartner, P. (1992), An ordered theory resolution calculus, *in* A. Voronkov, ed., 'Proceedings of the International Conference on Logic Programming and Automated Reasoning (LPAR'92)', Vol. 624 of *LNAI*, Springer Verlag, St. Petersburg, Russia, pp. 119–130.

Bibel, W. (1981), 'On matrices with connections', *Communications of the ACM* **26**, 633–645.

Boyer, R. S. (1971), Locking: A restriction of resolution, PhD thesis, University of Texas at Austin, Austin, TX.

Bronsard, F. & Reddy, U. S. (1992), Reduction techniques for first-order reasoning, *in* M. Rusinowitch & J.-L. Rémy, eds, 'Conditional Term Rewriting Systems, Third International Workshop', LNCS 656, Springer-Verlag, Pont-à-Mousson, France, pp. 242–256.

Bryant, R. E. (1992), 'Symbolic boolean manipulation with ordered binary-decision diagrams', *ACM Computing Surveys* **24**(3), 293–318.

Bürckert, H.-J. (1990), A resolution principle for clauses with constraints, *in* M. Stickel, ed., 'Proc. 10th CADE', Vol. 449 of *Lecture Notes in Computer Science*, pp. 178–192.

Davis, M. & Putnam, H. (1960), 'A computing procedure for quantification theory', *Journal ACM* **7**, 201–215.

de Nivelle, H. (1996), Ordering refinements of resolution, PhD thesis, Technische Universiteit Delft.

Dershowitz, N. (1987), 'Termination of rewriting', *J. Symbolic Computation* **3**, 69–116.

Eder, E. (1991), Consolution and its relation with resolution, *in* J. Mylopoulos & R. Reiter, eds, '12th International Joint Conference on Artificial Intelligence', Vol. 1, Morgan Kaufmann Publishers, Inc., San Mateo, CA, USA, Darling Harbour, Sydney, Australia, pp. 132–136.

Fermüller, C., Leitsch, A., Tammet, T. & Zamov, N. (1993), *Resolution Methods for the Decision Problem*, Vol. 679 of *Lecture Notes in Artificial Intelligence*, Springer-Verlag.

Fermüller, C. & Salzer, G. (1993), Ordered paramodulation and resolution as decision procedure, *in* 'Proc. LPAR'93', Vol. 698 of *Lecture Notes in Artificial Intelligence*, Springer-Verlag, Berlin, pp. 122–133.

Ganzinger, H. (1991), 'A completion procedure for conditional equations', *Journal of Symbolic Computation* **11**, 51–81.

Ganzinger, H., Meyer, C. & Weidenbach, C. (1997), Soft typing for ordered resolution, *in* 'Automated Deduction — CADE'14', Vol. ?? of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, pp. ??–?? To appear.
*http://www.mpi-sb.mpg.de/~hg/pca.html#CADE97

Ganzinger, H. & Nieuwenhuis, R. (1994), The Saturate system. User manual maintained on the Web.
*http://www.mpi-sb.mpg.de/SATURATE/Saturate.html

Hsiang, J. (1985), 'Refutational theorem proving using term-rewriting systems', *Artificial Intelligence* **25**, 255–300.

Huet, G. (1972), Constrained Resolution: A Complete Method for Higher Order Logic, PhD thesis, Case Western Reserve University.

Hustadt, U. (1997), Resolution-Based Decisison Procedures for Subclasses of First-Order Logic, PhD thesis, Fachbereich Informatik, Universität des Saarlandes, Germany. To appear.

Joyner Jr., W. (1976), 'Resolution strategies as decision procedures', *Journal of the Association for Computing Machinery* **23**(3), 398–417.

Kapur, D. & Narendran, P. (1985), An equational approach to theorem proving in first-order predicate calculus, *in* 'Proc. Ninth International Joint Conference on Artificial Intelligence', Los Angeles, CA, pp. 1146–1153.

Kirchner, C., Kirchner, H. & Rusinowitch, M. (1990), 'Deduction with symbolic constraints', *Revue Française d'Intelligence Artificielle* **4**(3), 9–52. Special issue on automatic deduction.

Klingenbeck, S. & Hähnle, R. (1994), Semantic tableaux with ordering restrictions, *in* A. Bundy, ed., 'Twelfth International Conference on Automated Deduction', LNAI 814, Springer-Verlag, Nancy, France, pp. 708–722.

Kowalski, R. A. (1974), Predicate logic as programming language, *in* 'Proceedings of the IFIP Congress', Amsterdam, Netherlands, pp. 569–4574.

Kowalski, R. A. & Kuehner, D. (1971), 'Linear resolution with selection function', *Artificial Intelligence* **2**, 227–260.

Lifschitz, V. (1989), 'What is the inverse method?', *Journal of Automated Reasoning* **5**(1), 1–24.

Loveland, D. W. (1969), 'A simplified format for the model-elimination theorem-proving procedure', *Journal ACM* **16**, 349–363.

Lynch, C. (1997), 'Oriented equational logic is complete', *Journal of Symbolic Computation* **23**(1), 23–45.

Manna, Z. & Waldinger, R. (1980), 'A deductive approach to program synthesis', *ACM Transactions on Programming Languages and Systems* **2**(1).

Maslov, S. J. (1964), 'An inverse method for establishing deducibility in the classical predicate calculus', *Dokl. Akad. Nauk SSSR* **159**, 1420–1424.

McAllester, D. A. (1993), 'Automated recognition of tractability in inference relations', *Journal ACM* **40**(2), 284–303.

Müller, J. & Socher-Ambrosius, R. (1988), Topics in completion theorem proving, Research Report Memo SEKI-SR-88-13, Univ. Kaiserslautern.

Murray, N. (1982), 'Completely non-clausal theorem proving', *Artificial Intelligence* **18**, 67–85.

Nieuwenhuis, R. & Rubio, A. (1992), Theorem proving with ordering constrained clauses, *in* 'Automated Deduction — CADE'11', Vol. 607 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, pp. 477–491.

Nieuwenhuis, R. & Rubio, A. (1994), AC-superposition with constraints: No AC-unifiers needed, *in* 'Proc. 12*th* International Conference on Automated Deduction', Vol. 814 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, pp. 545–559.

Nonnengart, A. (1995), A Resolution-Based Calculus for Temporal Logics, PhD thesis, Universität des Saarlandes, Saarbrücken, Germany.

Pais, J. & Peterson, G. (1991), 'Using forcing to prove completeness of resolution and paramodulation', *Journal of Symbolic Computation* **11**, 3–19.

Plaisted, D. & Greenbaum, S. (1986), 'A structure-preserving clause form translation', *JSC* **2**, 293–304.

Reynolds, J. C. (1965), Unpublished seminar notes. Stanford University.

Robinson, J. A. (1965*a*), 'Automatic deduction with hyper-resolution', *International J. of Computer Mathematics* **1**, 227–234.

Robinson, J. A. (1965*b*), 'A machine-oriented logic based on the resolution principle', *Journal of the ACM* **12**(1), 23–41.

Schmidt, R. A. (1997), Resolution is a decision procedure for many propositional modal logics, Research Report MPI-I-97-2-002, Max-Planck-Institut für Informatik, Saarbrücken, Germany. The extended abstract version is to appear in M. Kracht, M. de Rijke, H. Wansing and M. Zakharyaschev, editors, *Advances in Modal Logic '96*, CSLI Publications 1997.

Slagle, J. R. (1967), 'Automatic theorem proving with renamable and semantic resolution', *Journal of the ACM* **14**(4), 687–697.

Stickel, M. E. (1985), Automated deduction by theory resolution, *in* A. Joshi, ed., 'Proceedings of the 9th International Joint Conference on Artificial Intelligence', Morgan Kaufmann, Los Angeles, CA, pp. 1181–1186.

Tseitin, G. (1970), On the complexity of derivation in propositional calculus, *in* 'Seminars in Mathematics V.A. Steklov Math. Institute, Leningrad', Vol. 8, Consultants Bureau, New York-London, pp. 115–125. Reprinted in J. Siekmann and G. Wrightson (eds): Automation of Reasoning, Vol. 2, 1983, Springer-Verlag, pp. 466–486.

Vigneron, L. (1994), Associative-commutative dedution with constraints, *in* 'Proc. 12*th* International Conference on Automated Deduction', Vol. 814 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, pp. 530–544.

Wos, L. (1988), *Automated Reasoning: 33 Basic Research Problems*, Prentice-Hall, Englewood Cliffs, New Jersey.

Wos, L., Robinson, G. & Carson, D. (1965), 'Efficiency and completeness of the set of support strategy in theorem proving', *J. of the Association for Computing Machinery* **12**, 536–541.

Zhang, H. (1988), Reduction, superposition and induction: Automated reasoning in an equational logic, PhD thesis, Rensselaer Polytechnic Institute, Schenectady, New York.

Zhang, H. (1994), 'A new method for the boolean ring based theorem proving', *Journal of Symbolic Computation* **17**(2), 189–212.